
18

Man Page Listing

This chapter lists the man pages for backup and restore commands, volume management commands, media duplication command, and HSM commands. You can display a full description of each command by typing **man** followed by the command name.

The following three categories of man pages are described:

- Backup and Restore Man Pages
- Volume Management Man Pages
- HSM Man Pages

Backup and Restore Man Pages

This section describes backup and restore commands that can be run from the EDM server. These commands can be used, among other things, to initiate backups (**ebbackup**) and restores (**ebrestore** and **ebcrecover**), start the EDM window (**edm** and **edmrestore**), and generate a variety of reports (**ebreport** and **ebcreport**) directly from the EDM server's command line.

Backup and Restore Commands and Daemons	Description
eb	Introduces the EDM Backup product, programs, daemons, and man pages.
eb_build_htab	Updates the database of hosts known to the EDM Backup Client Installation wizard.
eb_dc_backup	Starts an EDM Symmetrix Connect backup.
eb_dc_restore	Restores data backed up via a direct connect backup.
eb_deinstall_client	Deinstalls the EDM Backup client software using the command-line interface.
eb_deinstall_server	Deinstalls the EDM Backup server software using the command-line interface.
eb_install_client	Installs the EDM Backup client software (called by the EDM Backup Client Installation Wizard).
eb_install_server	Configures the EDM Backup server software (called by eb_server_config).
eb_rehome_client	Updates an EDM Backup client to know that it has a different EDM Backup server than the one from which it was originally installed.
eb_server_config	Configures a host for use as a backup server or deconfigures it.
eb_sybconf_db	Configures online Sybase database backups.
ebbackup	Initiates and controls client backups under EDM Backup.

Backup and Restore Commands and Daemons	Description (Continued)
ebbbackupd	Performs a backup of a single client system under EDM Backup.
ebcatalogd	Supervises the post-processing of backup catalogs.
ebcatclean	Deletes backups, catalogs, and saveset records that expired or are unreferenced under EDM Backup.
ebcatcomp	Completes the information stored in a backup catalog file and creates backup catalog deltas.
ebcatproc	Forces the processing of a backup catalog.
ebcatsort	Sorts an EDM Backup catalog file.
ebcp	Copies data from one place to another on an EDM Backup server, an EDM Migration client, or between two such machines.
ebcrecover	Provides an easy way to execute the ebrestore command from an EDM Backup client using the client's native connection method. (It actually calls ebrecover on the EDM, which is just a symbolic link to the actual restore program, ebrestore .)
ebcreport	Enables users to run ebreport from an EDM Backup client using the client's native connection method.
ebexpire	Deletes backup data, saveset records, and backup catalogs that have expired under the EDM Backup system.
ebfs_dump_vol	Reads a volume of backup media producing a hexadecimal listing of its contents an/or an extended-cpio stream that can be read by recxcpio to restore a directory hierarchy.
ebimport	Imports backup media, backup catalogs, and saveset records.
eblistend	Listens for requests from DBMS clients for backups and restores, and starts them as needed.
ebrestore	Restores backup files from backups created by the EDM Backup system.
ebreport	Produces several backup reports including media, history, and disaster reports. (See Chapter 16 "Backup Reports and Log Files".)

Backup and Restore Commands and Daemons	Description (Continued)
ebtreegen	Generates a tree index for a backup catalog file. (Evoked by ebcatalogd .)
edm	Starts the EMC Data Manager graphical user interface.
edmcrestore	Starts the EDM Restore window and displays it on the backup client.
edmhlp	Starts the online Help facility for the EDM.
edmlinkq	Allows you to query the remote client via the EDM Transfer Protocol to provide version and capability information. This has the side-effect of allowing the client-server connection to be tested.
edmproc	Lists, starts up, shuts down or restarts all edm daemon processes. edmproc performs the start up/shut down operations in the correct order.
edmremote	Starts the EDM GUI on a remote EDM server and displays it on a specified host.
edmreport	Executes saved reports on currently running or completed backups on a local EDM or on an EDM Domain. If using a domain, the EDM Domain Master machine must be a trusted machine, because the local administrator can change passwords and run the edmreport CLI without knowing the Domain Login Credentials.
edmrestore	Starts the EDM Restore window on the EDM Backup server.
epcleanup	Removes files that are no longer needed. Usually run from crontab.
epcomm_util	Command line utility for communicating with the EDM Client Communications Daemon, epcommmd.
epnewlog	Rotates, archives, or truncates system logs. Usually run from crontab.
epshowmod	Displays all or selected installed EDM modules.
epshowpath	Displays the installation location of EDM software.
epshowprod	Displays information for a selected or all installed products.

Backup and Restore Commands and Daemons	Description (Continued)
epshowvers	Displays all products and their versions available on the EDM distribution CD.
eptrunclog	Truncates the daily message log file and mails a copy to specified users. This is usually run from crontab.
findxcpio	EDM Backup's client find and xcpio program.
ntexchreport	Reports on the Microsoft Exchange backups that have been made to an EDM server.
ntexchrestore	Restores a Microsoft Exchange object that was backed up by the EDM server.
ntsqlreport	Reports on Microsoft SQL Server database backups made to the EDM server.
ntsqlrestore	Restores a Microsoft SQL Server object that was backed up by the EDM.
olddb_exec	Initiates an Oracle Online backup or restore from the EDM.
portservices	Modifies the edm_services files on the EDM, enabling, changing, and disabling port control settings.
rasd	RASD (Reliability Agent Scanner daemon) monitors significant system events that inhibit the successful completion of EDM applications.
recxcpio	Creates a directory hierarchy that corresponds to the contents of an extended-cpio stream, such as that produced by findxcpio and the ebrestore program.
snmpeved	The SNMP subagent for Volume Management and EDM Backup.
sybrecover	Recovers striped database back ups from the EDM to a Sybase server.
sybreport	Reports on Sybase backups made to the EDM server.

Volume Management Man Pages

The Volume Management command line interface (CLI) enables you to monitor volume management activities and perform administrative tasks at the command line level. You can use the CLI in a non X windowing environment (such as dial-in-sites). The CLI also enable you to write shell scripts to automate volume management tasks.

Volume Management Commands and Daemons	Description
dbreport	Generates volume reports from the volume database.
edmlm	Starts the graphical EDM Library Unit Manager separate from the EDM GUI..
evmaddtempl	Creates a volume template for a specified media type and application.
evmchtempl	Modifies a volume template for a specified media type and application. This can be used to increase maximum usage of a template.
evmchvol	Changes the attribute settings for an existing labeled volume. This can be used to increase maximum usage of a volume.
evmclean	Cleans drive(s) in a named library unit. Before cleaning a drive, the cleaning cartridge must be present in the library unit. (See Description .)
evmctl	Queries or sets attributes of the EDM Volume Management system or individual library units.
evmeject	Ejects the specified volume or cleaning cartridge from a library unit. This works only with library units that have an inlet. (See Description to get a volumes identifier.)
evmenable	Enables a drive or library unit. Volume management disables a drive when certain errors occur.
evmimport	Imports one or more volumes into a server's volume catalog. Use this command when you move volumes from one server to another or to reconstruct a volume catalog that was destroyed.
evminject	Inserts a volume into a library unit. (See Description .)

Volume Management Commands and Daemons	Description (Continued)
evminventory	Initiates an inventory of the volumes in the named library unit.
evmlabel	Labels the selected volume by using a specified volume allocation template. Use Description -t to list the names and IDs of all defined templates and Description to view a template's attributes.
evmlistd	Lists all running volume management processes.
evmmount	Queues a mount request for the specified volume. Generally, volume management handles all mount and dismount requests. Therefore, this command is intended only for mounting foreign volumes.
evmreject	Rejects a queued request for a volume.
evmrmtmpl	Removes the specified volume template from the volume management template database.
evmrmmvol	Removes all knowledge of an existing volume from a server's volume catalog.
evmseterror	Enables you to set the error and/or warning count for a drive, volume, or library unit.
evmstat	Provides status of volume management system, including devices (library units and drives), volumes, media types, and system notifications.
evmtmpl	Displays attributes of one or more volume templates.
evmumount	Removes a volume from a drive that was mounted by using evmmount .
evmvol	Displays attributes of one or more volumes.
evmwhere	Describes the attributes that are obtained by running evmvol -V . This is NOT a command. This man page describes the where -clause syntax only.
lmconfig	Configures library managers and device drivers for the EDM.

Volume Management Commands and Daemons	Description (Continued)
vmdaemon	EDM volume management daemon. Note: This command should not be run from the CLI.
vmdup	Manages media duplication for specified trails or volumes by turning on duplication for manually-duplicated trails. This is also used to reschedule duplication for a given volume, in case a duplication failed.
vmdupcfg	Displays the current values of the duplication configuration parameters, and allows the values to be changed.
vmdupd	Run from the command line to alter the state of the currently-running vmdupd daemon that controls the media duplication, as well as to display the list of volumes currently scheduled for duplication. Only one occurrence of the vmdupd (run without arguments) can run at a time.

HSM Man Pages

This section describes the following types of migration commands: staging and staging configuration commands, network migration server commands, and user level commands.

Most users do not need the user level commands. These are useful for those who set up application environments and for those who need to understand filesystem usage patterns.

User level commands are marked with an asterisk (*) in the following table.

HSM Commands and Daemons	Description
ebcheck	Finds files that have inconsistencies with their staging IDs and fixes them.
em_new_volume	Allocates a new staging volume for a specific staging template.
embsi *	Stages specified files in from staging media or client stores, which ensures that they are completely resident on magnetic storage.
emcheck *	Checks the HSM client and server configuration to verify its correctness, warns you of potential problems, and corrects inconsistencies.
emchmod *	Sets the staging control properties for a file or directory. Unlike chmod , it clears unspecified properties.
emcompact	Automatically compacts staging volumes. Usually run from crontab.
emcrecover_wait *	Waits for a set of client store bitfiles to be restored on an EDM with HSM. It scans the set of files listed on the command line and checks the status of every bitfile referenced by this set.
emcreport *	Displays information about client store usage and identifies the current staging targets for each stageable filesystem.
emdu *	Displays the number of KB contained in all files and directories specified. Using the -v option displays the amount of virtual space, which includes the space on the EDM if the files have a staging image; otherwise it is the space on the local magnetic disk.

HSM Commands and Daemons	Description (Continued)
emfind *	Recursively descends the directory hierarchy for each pathname in the pathname-list, seeking files that match a logical expression. Supports several additional predicates over find .
emfsconf	Configures HSM filesystems by assigning filesystems to staging templates, removing filesystems from staging templates, and changing filesystem parameters.
emfsdeconfig	Deconfigures a migration filesystem.
emfsreport *	Produces virtual filesystem statistics. It displays the amount of stageable filesystem data, and the amount of data that is currently staged. It displays your working set in days of usage.
emls *	Lists file attributes. Similar to ls , but lists staging attributes, including the number of KB on magnetic disk, the number of KB currently staged out, and server and client store if any, to which the file is staged, as well as the staging control properties, such as residence priority.
emlsconf	Displays the current staging configuration parameters.
emscheck	Checks the network migration server configuration files. The emscheck command checks all global and store specific files for syntactic and semantic correctness. In addition, it performs certain clean up operations on client stores. Run emscheck nightly via cron .
emschs	Changes certain configuration parameters for an existing store.
emsconfig	Changes the protocol interface parameters. Only use at the direction of your customer service organization.
emsd	The network migration server daemon. The emsstart command always invokes this daemon during system startup. The daemon handles all network client HSM protocol requests from the client systems.
emsdefs	Changes certain default store configuration parameters. The emsdefs command affects the operation of the emsmks command.
emshalt	Terminates the network migration server. The emshalt command kills the running emsd processes and aborts any outstanding staging operations.

HSM Commands and Daemons	Description (Continued)
emsinit	Initializes the network migration server configuration files. If no configuration files exist, emsinit creates an initial configuration with no stores and standard default definitions. The emsinit command is normally only run during installation.
emslss	Lists all configured stores, or individual stores selected by name or owning client.
emsmks	Makes a client store. You can also use emsmks to insert an existing store tree into a server configuration database.
emsmvs	Renames and/or moves an existing store.
emsrms	Removes an existing store.
emsstart	Initiates the network migration server. This has no effect if the network migration server is already running. The emsstart command is run during system startup.
emsstat	Displays network migration server usage statistics. It displays both cumulative and incremental statistics. Use emsstat to determine the current status of network migration services.
emstage *	Explicitly stages out the specified files. You must be the file owner or the superuser to use this command.
emstconf	Creates new staging templates, removes existing staging templates, and changes parameters for existing staging templates.
emsundel	Starts a restore run to retrieve bitfiles listed in the recover_list files. Only run emsundel when an operator is available to handle volume mount requests.
emsysconf	Sets system-wide staging parameters in the configuration database.
emvck	Checks and corrects staging volume statistics. This is usually run from crontab.
restage	Stages or restages files to the specified staging trail.

Part V Disaster Recovery

19

Being Prepared for a System Disaster

If there is a disaster, you should be prepared for a disaster recovery. However, feel free to call Customer Service with questions.

CAUTION: Do not wait for a disaster to read this chapter. The information in this chapter is about steps that must be taken with each backup so that you will be able to recover when a disaster occurs.

This chapter tells you how to prepare for the disaster recovery of your backup files. The following two chapters contain overall disaster recovery procedures to use when you experience a disk crash on an EDM server or client. Because each disaster is unique, these steps are presented only as a guide and not as all-inclusive, step-by-step instructions.

CAUTION: Performing a disaster recovery requires experience with EDM Backup administration (and HSM administration, if you have the HSM option), UNIX system administration, and the site environment.

You must establish a disaster strategy and safeguard your media and the reports *before* a disaster occurs. Otherwise, you will not have the necessary information to recover your system to its original state.

Note: You should develop a disaster recovery plan that meets your specific organizational goals. Actual file recovery can be as simple as off-site tapes, or as complex as duplicate Symmetrix systems utilizing SRDF/Timefinder.

To fully recover from a system disaster, you must run regular backups, safeguard your backup media, and run and save the appropriate backup reports. You can provide additional protection by creating redundant media and storing it offsite.

Safeguarding Your Backup Media

To be prepared for a system disaster, you must run regular backups and save the backup media for both the current and previous full rotation periods. For example, assume the following:

- The rotation period is 7 days.
- A full backup is run on Monday.
- Every backup generates a single new piece of media.

If today is Tuesday, you need the media that was generated since Monday of the previous week, or the last eight pieces of media. Tomorrow, you will need the same eight pieces, plus the media that is generated from today's backup.

Be sure to save the backup media in a safe place, preferably offsite or onsite in a fireproof vault.

CAUTION: Failure to have visually identifiable labels on removable media will significantly complicate and lengthen the Disaster Recovery procedure.

Be sure removable backup media is physically labeled (such as with barcodes) so it can be visually identified if needed during the disaster recovery process. If barcoding is not used, each piece of media must be physically labeled with its assigned sequence number.

Running and Saving Reports

At the end of your backups for the day, your LOCAL_DATABASE is automatically backed up, which provides you an exact picture of the EDM Backup database.

At the completion of every LOCAL_DATABASE backup, the /usr/epoch/EB/config/local_db_cleanup script automatically generates a MINIMAL Disaster Report. By default, this report is emailed to all EDM Backup administrators, appended to /usr/epoch/EB/config/disaster-report.log, and printed to the default system printer.

CAUTION: It is essential that you save a hard or soft copy of the MINIMAL Disaster Report after each backup. Keep it in a fireproof location, either offsite or in an onsite fireproof vault.

If the LOCAL_DATABASE work item remains in the schedule for more than 24 hours without being run, it will be forced to run immediately. This is known as a “late” LOCAL_DATABASE backup. The work item remains in the schedule to be run again normally, or forced if needed.

ebbackup displays a message and **ebreport disaster** notes a “late” LOCAL_DATABASE backup.

MINIMAL Disaster Report

This MINIMAL Disaster Report is a subset of the FULL Disaster Report that **ebreport disaster generates**. It provides essential information that you need to perform a disaster recovery on the server—a list of media volumes for the most recent LOCAL_DATABASE backup, the current EDM Backup configuration, the current Library Manager configuration, copies of the key configuration-file settings, and information about baseline backups.

Note: This MINIMAL Disaster Report does *not* include backup client information.

FULL Disaster Report

You should run the FULL Disaster Report once every backup rotation and whenever significant system changes are made. The following example runs the FULL Disaster Report and redirects it to a file:

```
emc# ebreport disaster >  
~sysadmin/disreports/960917
```

See “Backup Disaster Reports” on page 16-19 for a description of the FULL Disaster Report.

Redundant Backup Coverage

In preparation for a possible disaster, it is recommended to have redundant backup coverage so that you can move some backup media offsite for safe keeping. Following are two ways of providing redundant backups.

Configure Alternate Media Sets (Trailsets)

One good backup strategy is to configure an alternate media set (trailset) for use on alternate nights. (A *trailset* contains all of the media that is used in performing full and incremental backups for a backup schedule template in a single rotation period.)

For example, with a rotation period of seven days:

- with a primary trailset only, a complete trailset includes at least one full backup and six incrementals for each work item.
- with an alternate trailset, each complete trailset includes at least one full backup for each work item, but only two or three incrementals.

With primary trailsets only, move backup media offsite as soon as it is older than one rotation period. With alternate trailsets, you can move each backup volume from one trailset offsite as soon as the volume is full.

Media Duplication

Another option is to use Media Duplication, which enables you to create a duplicate set of backup media automatically after each backup session.

After you configure media duplication in the EDM Backup Configuration window, the duplication of a set of backup media occurs automatically after each backup session. You can then take the duplicate media offsite for safekeeping.

This method does not use network bandwidth and can be a good choice if you have extra drives. For more information, see Chapter 9 “Media Duplication”.

20 Recovering a Server from a Disk Failure

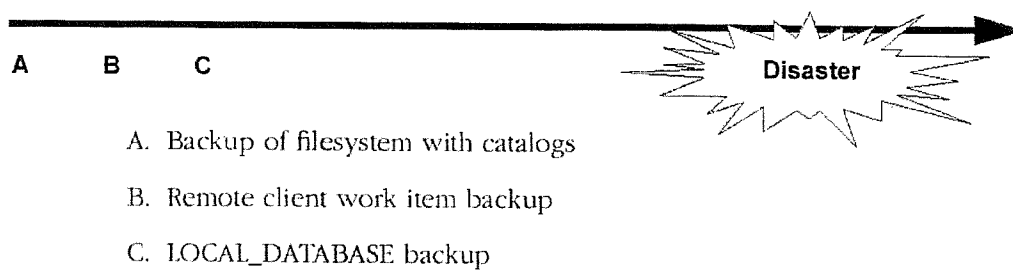
This chapter contains disaster recovery procedures to use when you experience a disk crash on a backup server. Because each disaster is unique, these steps are presented only as a guide and not as all-inclusive, step-by-step instructions.

CAUTION: Performing a disaster recovery requires experience with EDM Backup and HSM (optional) administration, UNIX system administration, and the site environment.

To restore a backup server, determine the extent of the damage and disable backups, replace any damaged disks, reinstall the operating system, and reinstall any lost EDM Backup or HSM software.

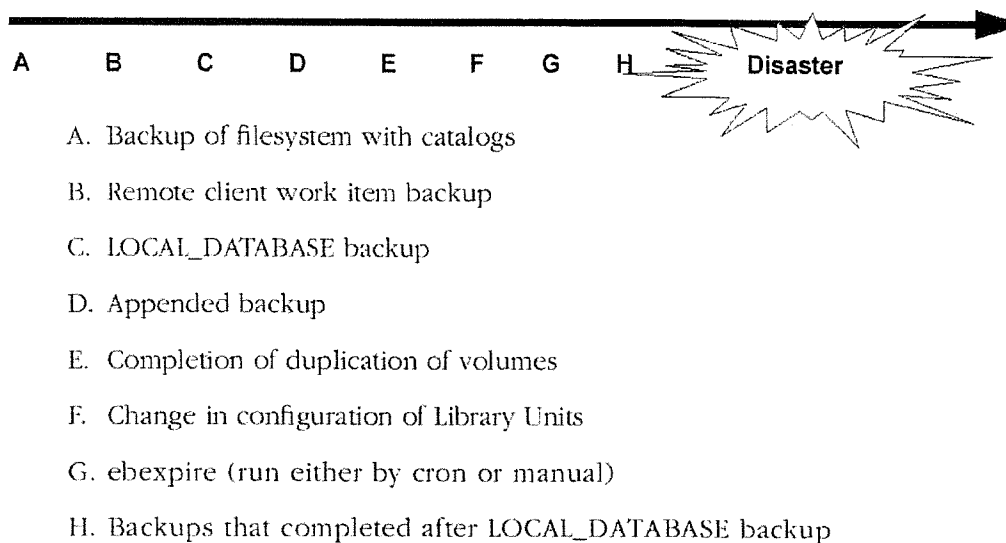
Prior to restoring lost files, you import the volume management and backup catalog database information and then use EDM Backup to restore the databases as they were at the time of the last LOCAL_DATABASE backup. After doing this, you can restore all of the data from changes that occurred after the last LOCAL_DATABASE backup.

The procedures in this chapter are based upon the example in Figure 20-1 that shows the actions that occurred before the disaster in the simplest case. Figure 20-2 illustrates these and other actions that may have occurred before the disaster.

Figure 20-1**Disaster Immediately after the LOCAL_DATABASE Backup**

Your disaster could occur immediately after your last LOCAL_DATABASE backup (Step C, above). If so, follow all of the sections except those that are marked "For changes after the last LOCAL_DATABASE backup."

Figure 20-2

Disaster after Other Possible Actions

Events, such as D, E, F, G and H in Figure 20-2, may have occurred (in any order) after the last LOCAL_DATABASE backup. If your disaster occurs after one or more of these, read ALL sections in the chapter, including those that are marked “For changes after the last LOCAL_DATABASE backup,” and follow those that fit your situation.

Steps to Restore a Server

The actions that are required to restore a server from a disk failure are discussed in this chapter. Follow the steps below to restore a backup server in case of a disk crash or major loss of files. Your situation may vary from this example and you may not need all of these sections. If some filesystems remain intact, you may be able to skip some steps. You should adjust your disaster recovery steps accordingly.

You need information from the Disaster Report for many of these steps. For information about this report, see “Running and Saving Reports” on page 19-3.

Note: The procedures that follow use *emc* as the name of the server. Substitute the name of your own server in each of the examples.

Each of the following steps are described in the following sections of this chapter. Perform them in the order given.

1. Stop All Activity on the Server
2. Reinstall Hardware and Software as Needed
3. Temporarily Reconfigure the Server
4. Restore LOCAL_DATABASE Files

The next two steps are only done if you made changes after the last LOCAL_DATABASE backup.

5. Reconfigure Library Units
6. Restore Catalogs and Backup Information Created After LOCAL_DATABASE Backup

Follow these steps for all disaster recoveries.

7. Restore Data Created Before LOCAL_DATABASE Backup
8. Reenable crontab Entries
9. Restore Past Catalogs
10. Restore Missing Catalogs

Stop All Activity on the Server

Because you perform all of the restore in multiuser mode, you must notify the user community to stop all activity on the server. No one should be able to log in. The server must be inactive before you perform any disaster recovery procedures.

Disable Activity

Edit root's crontab file (/var/spool/cron/crontabs/root) on the server to comment out all backup and HSM staging commands that may start up automatically. Following are commands to comment out:

```
00 0 * * * /bin/kill -1 `cat /usr/epoch/etc/mal/emmasterd.pid` >/dev/null
2>&1#EPCmalc
15 23 * * * /usr/epoch/bin/emvck >/dev/null 2>&1#EPCmalc
00 1 * * * /usr/epoch/bin/emcompact -c >/dev/null 2>&1#EPCmalc
00 2 * * * /usr/epoch/bin/emscheck >/dev/null 2>&1#EPCmalc
00 3 * * * /usr/epoch/bin/emsundel >/dev/null 2>&1#EPCmalc
00 18 * * * /usr/epoch/EB/bin/ebbackup default >/dev/null 2>&1 #EPCebs
00 11 * * * /usr/epoch/EB/bin/ebexpire -expire -purge >/dev/null 2>&1 #EPCebs
00 1 * * * /usr/epoch/EB/bin/ebcatclean -fix_saveset >/dev/null 2>&1 #EPCebs
00 3 * * * /usr/epoch/EB/config/local_db_warning >/dev/null 2>&1 #EPCebs
```

Reinstall Hardware and Software as Needed

Before you perform a restore, you must have your hardware and software in the same condition as that before the disaster occurred.

Determine the extent of the damage to both hardware and software. Be sure to identify and replace all damaged hardware *before* restoring any software.

1. Check hardware and replace if necessary. You can use the disk diagnostics capability of a program such as **format** to determine whether anything is wrong with your disk.

Make sure the replacement hardware is fully compatible with the system you had before the disaster. Each new disk should have at least the same storage capacity as the old disk.

CAUTION: Do not continue with software recovery until you are absolutely certain that the disks are free of hardware problems.

2. Check the filesystems for loss of software, starting with the operating system.

If `/` or `/usr` was destroyed, reinstall the server's native operating system following the platform-specific instructions that are provided with your server. This partitions your disks, restores the root filesystem, and restores the portion of `/usr` that the operating system loads.

To partition the disks correctly, use the disk configuration data from the last Disaster Report (which is described in "Running and Saving Reports" on page 19-3) See Figure 20-3 on page 20-7. Make sure you rebuild these filesystems to the same size (or larger) and the same number of inodes (or more) as they were before the disaster occurred.

Figure 20-3

Locally Mounted Disks from the Disaster Report

Displaying locally mounted disks...

```

/ (/dev/dsk/c0t3d0s0 ):      8192 block size   1024 frag size
230302 total blocks  92988 free blocks  69968 available  60416 total file
56761 free files      8388632 filesys id
   ufs fstype        0x00000004 flag           255 filename length

/usr (/dev/dsk/c0t3d0s3 ):    8192 block size   1024 frag size
673742 total blocks  295032 free blocks  227672 available  169920 total f
153786 free files      8388635 filesys id
   ufs fstype        0x00000004 flag           255 filename length
. . .
. . .
/ep_usr (/dev/dsk/c0t3d0s4 ):  8192 block size   1024 frag size
774766 total blocks  396532 free blocks  319072 available  196352 total f
193845 free files      8388636 filesys id
   ufs fstype        0x00000004 flag           255 filename length
. . .
. . .

```

3. If you have an HSM system, you need to reinstall VERITAS VxFS. See your *VERITAS File System (VxFS) Quick Start Guide*.

Note: If you are reinstalling the operating system, you need a VxFS license.

4. Verify whether the filesystem that contains the backup software is damaged or missing (the EDM Software Filesystem, /ep_usr in the example in Figure 20-4).
5. Check the Disaster Report to see what you had at the time of the last report.

Figure 20-4**Displaying /etc/vfstab... from the Disaster Report**

Displaying /etc/vfstab...

Oct 3 14:23 1998 /etc/vfstab Page 1

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
#						
fd -	/dev/fd fd	-	no	-		
/proc -	/proc proc	-	no	-		
/dev/dsk/c0t3d0s1	-	-	swap	-	no	-
/dev/dsk/c0t3d0s0	/dev/rdisk/c0t3d0s0			ufs	1	no -
/dev/dsk/c0t3d0s3	/dev/rdisk/c0t3d0s3	/usr		ufs	1	no -
/dev/dsk/c0t3d0s4	/dev/rdisk/c0t3d0s4	/ep_usr		ufs	2	yes -
/dev/dsk/c0t1d0s2	/dev/rdisk/c0t1d0s2	/data		vxfs	2	yes -
/dev/dsk/c0t5d0s0	/dev/rdisk/c0t5d0s0	/data1		vxfs	2	yes -
/dev/dsk/c0t5d0s1	/dev/rdisk/c0t5d0s1	/data2		vxfs	2	yes -
/dev/dsk/c0t5d0s3	/dev/rdisk/c0t5d0s3	/data5		vxfs	2	yes -
/dev/dsk/c0t2d0s0	/dev/rdisk/c0t2d0s0	/data6		vxfs	2	yes -
/dev/dsk/c0t2d0s1	/dev/rdisk/c0t2d0s1	/data7		vxfs	2	yes -
/dev/dsk/c0t2d0s3	/dev/rdisk/c0t2d0s3	/data8		vxfs	2	yes -
/dev/dsk/c0t0d0s0	/dev/rdisk/c0t0d0s0	/data3		vxfs	2	yes -
/dev/dsk/c0t0d0s1	/dev/rdisk/c0t0d0s1	/data4		vxfs	2	yes -
swap -	/tmp	tmpfs-	yes	-		

6. Use the Installation Report portion of the Disaster Report to determine which of the various binaries were actually in /usr and which were symbolically linked to another filesystem. The EpochBackup Installation Report section shows the directories and symbolic links for the backup software.

In the third line of the split-install example shown in Figure 20-5 on page 20-9, the backup software is installed in /ep_usr/epoch and a symbolic link is in /usr/epoch and points to /ep_usr/epoch.

Also, in this example, catalogs, db, and log are installed in /data/epoch/EB/ and a symbolic link is in /usr/epoch/EB and points to them.

Figure 20-5**Installation Report in the Disaster Report**

```
EpochBackup Installation Report for server adam at Oct 13 09:01:14 1998
Report options: -all

Installed Software:

EDM High-Performance Centralized Backup with HSM
  abbreviation:  edmhsm
  version:      4.0.0.5
  server platform:  sun4_5.5.1
  platform:     sun4_5.5.1
  patch id:     none
  install date:  19971110165742
  modules:      EPCql EPCtps EPCsnmp EPCesl EPCdevlib EPCdev EPCelmlib EPCelm
EPCebhslb EPChsesdm EPCmalib EPCmalc EPCmawrp EPCebsedm EPCebc EPCgui
EPColdoc

  status:      complete

(Installation Report continued on next page.)
```

(Installation Report Continued from previous page.)

Oracle Platinum On-Line Database Backup

```

abbreviation:  oraplt
version:       1.1.0.5
server platform: sun4_5.5.1
platform:     sun4_5.5.1
patch id:     none
install date:  19971112104243
modules:      EPCoraplt
status:       complete

```

EpochBackup currently running load 7.0.0.0

```

/usr/epoch IS A SYMLINK to /ep_usr/epoch
/usr/epoch/EB is a real directory under /ep_usr/epoch
/usr/epoch/GENDIR IS A SYMLINK to /home/epoch

```

Backup
Software

```

/usr/epoch/EB/adam is a real directory under /usr/epoch/EB
/usr/epoch/EB/bin is a real directory under /usr/epoch/EB
/usr/epoch/EB/catalogs IS A SYMLINK to /home/epoch/EB/catalogs
/usr/epoch/EB/client is a real directory under /usr/epoch/EB
/usr/epoch/EB/config is a real directory under /usr/epoch/EB
/usr/epoch/EB/db is a real directory under /usr/epoch/EB
/usr/epoch/EB/locks is a real directory under /usr/epoch/EB
/usr/epoch/EB/log IS A SYMLINK to /home/epoch/EB/log
/usr/epoch/EB/preconfig is a real directory under /usr/epoch/EB
/usr/epoch/EB/tmp is a real directory under /usr/epoch/EB

```

Backup
Software
Directories

The local client is of the type: sun_sun4_v55_srv

The client backup username is: ebadmin

The user ID for ebadmin is: 24375

The group ID for ebadmin is: 25

The home directory for ebadmin is: /usr/epoch/EB

```

Client chip is of type ibm_rs6000_v325 (6.0.0.0), installed Tue May 20 14:05:23 199
Client yyz is of type windows_nt_all (6.0.0.0), installed Thu Sep 11 15:04:42 199
Client perf-prol is of type windows_nt_all (5.0.1.0), installed Thu Sep 25 15:58 199
Client pilgrim is of type sun_sun4_v54 (6.0.0.0), installed Thu Oct 23 15:53:58 199
Client adam is of type sun_sun4_v55_srv (6.0.0.0), installed Wed Nov 12 11:03:07 199

```

End of EpochBackup Installation Report for server adam at Mar 13 09:01:14 1998

7. Reinstall the backup software.

Use the data from the Installation Report section of the Disaster Report to determine where and how it had been installed most recently (see Figure 20-5 on page 20-9).

- a. If the filesystem that contained the backup software was destroyed, reinstall the software as a scratch installation. See your *Software Installation* manual.
- b. If the filesystem that contained the backup software is still intact but /usr was destroyed, you need to recreate the symbolic link from /usr that points to this filesystem, which contains the backup software.

To match the example in Figure 20-5 on page 20-9, enter the following (specify your own installation directory):

```
emc# ln -s /ep_usr/epoch /usr/epoch
```

Temporarily Reconfigure the Server

You must configure the library unit by using **lmconfig**, and then a temporary backup configuration by using **eb_server_config**.

lmconfig

1. Obtain the **lmconfig** data from the Disaster Report (see Figure 20-6).

Figure 20-6**Library Manager Configuration from the Disaster Report**

Displaying library manager configuration (used with lmconfig...)

* Lu_name	Name	ID	Status
L offline_0	-	-	synced
L offsite_0	-	-	synced
L at_dlt_3264_0	-	(0,1,1,0)	synced
D at_dlt_3264_0	drive_0	(0,1,5,0)	enabled
D at_dlt_3264_0	drive_1	(0,1,4,0)	enabled
D at_dlt_3264_0	drive_2	(0,1,3,0)	enabled
L hp_mf_c17xx_0	-	(0,2,6,0)	synced
D hp_mf_c17xx_0	drive_0	(0,2,5,0)	enabled
D hp_mf_c17xx_0	drive_1	(0,2,4,0)	enabled
D hp_mf_c17xx_0	drive_2	(0,2,2,0)	enabled
D hp_mf_c17xx_0	drive_3	(0,2,1,0)	enabled

2. Remove all of the media from the library unit that you just configured and put the media aside, in their own box or some other location where they cannot be confused with other media. You reuse these media when you restore the catalogs (see page 20-23).

This significantly shortens the time that is required to inventory this library unit when you reboot the server.

3. Using **lmconfig**, install the drivers for a library unit that supports the media you use later to import and restore the LOCAL_DATABASE backup.

See Chapter 17 “Configuring Library Managers” for details about running **lmconfig**.

Note: You can save a significant time by configuring only one library unit, even if more than one is available.

4. Use the **CONFIG** option of **lmconfig** to configure the library unit.

Note: If you use the **AUTOCONFIG** option, you must leave at least one piece of media in the TLU. If you do this, leave the piece of media required for LOCAL_DATABASE restore.

5. Using the LOCAL_DATABASE volumes section of the Disaster Report (see Figure 20-7 on page 20-16), reinsert the volumes that contains the LOCAL_DATABASE backup into the library unit that you configured.
6. Reboot the server again with the following command:
emc# **/usr/sbin/shutdown -y -i6 -g0**

The Library Manager performs an inventory of this library unit. When the inventory is complete, continue to the next step.

eb_server_config

Configure a temporary EDM Backup configuration. The original EDM Backup configuration is restored later, when you restore the LOCAL_DATABASE files as described in the next section.

1. Determine whether the backup server software was a split directory or single directory server configuration.

Look at the Installation Report in the Disaster Report (see Figure 20-5 on page 20-9).

- a. If catalogs, db, and log are real directories, not SYMLINKs, you have a single directory installation.

- b. If either catalogs, db, or log are SYMLINKs to another directory, you have a split directory installation.
- 2. Run **eb_server_config** and respond to the prompts to recreate the installation and configuration of the server, as indicated in the Installation Report.
 - a. If the previous installation was a single directory installation, you need to select *no* when asked if you wish to install into a “split” directory.
 - b. If the previous installation was a split directory installation, you need to select *yes* when asked if you wish to install into a “split” directory.
 - c. Determine the parent directory of the catalog, db, and log directories (in Figure 20-5 on page 20-9) that would be /usr/epoch/EB, and enter that path when asked for the name of the target directory.
 - d. Determine the client backup username by examining the Installation Report of the Disaster Report (see Figure 20-5 on page 20-9), and enter it when asked.
 - e. Answer the remaining questions as you did in the original installation and as shown in the Disaster Report.

Restore LOCAL_DATABASE Files

Once all hardware and software are in place, and you temporarily reconfigured the server, you need to restore your most recent LOCAL_DATABASE backup.

1. Import the original or the duplicate volumes that contain the LOCAL_DATABASE backup (which was inserted at step 5 on page 20-13) into the Volume Manager catalog. To do this, enter:

```
emc# evmimport -l LibraryUnitName -a
```

where *LibraryUnitName* is the name of the library unit you configured earlier in the procedure. The **-a** option causes all of the volumes in that library unit to be imported.

2. Use the LOCAL_DATABASE section of the Disaster Report (see Figure 20-7) to get the saveset ID and first volume ID of the LOCAL_DATABASE backup.

Note: If you plan to use duplicate volumes, you only need to load the duplicate volumes in the library unit. If the original volume is in the offline state, the substitution of duplicates happens automatically. Always use the original volume ID when needed in the CLI commands or in the ebimport CLI, do not use the duplicate volume ID.

Figure 20-7

LOCAL_DATABASE Section of the Disaster Report

Saveset ID	Original Volume ID	Duplicate Volume ID
EDM Backup MINIMAL Disaster Report for server "elf" on May 18 13:05:42 1999		
LOCAL_DATABASE Backup Volumes Report		
The following volumes contain the most recent LOCAL_DATABASE backup which will be required in the event of a Disaster Recovery:		
Saveset ID 7271A0D4.37408006 for LOCAL_DATABASE backup on 5/17/99 16:45		
	backup_DLT #0010 (38DD01FC8C977D94) [original] - currently in library unit "de_dlt_x700_0", slot #5	
	backup_DLT #0011 (EBDD020907E79824) [duplicate] - currently in library unit "de_dlt_x700_0", slot #3	
This LOCAL_DATABASE backup will require 27.2 MB of disk space to be recovered		

3. Import the LOCAL_DATABASE backup to restore the saveset records and catalogs.

Run **ebimport** using the **-media** option (with the 16-character original volume ID listed in parentheses for the LOCAL_DATABASE backup volume) and the 17-character LOCAL_DATABASE saveset ID, as follows:

```
emc# ebimport -media 38DD01FC8C977D94 7271A0D4.37408006
                        |                      |
                        Original Volume ID      Saveset ID
```

At this point, enough data is restored to restore the LOCAL_DATABASE backup.

4. Make sure catalog processing completed. To do this, enter:
emc# **ebcatalogd -status**

When it reports that all catalogs are processed, continue to the next step.

5. If you have an HSM system, reconfigure all migration controlled filesystems that must be restored.

Refer to **emlsconf** output (see Figure 20-8 on page 20-18) in the Disaster Report and use exactly the staging parameters that it reports as arguments to the **emsysconf**, **emstconf**, and **emfsconf** commands. Valuable control information was lost when the filesystems were damaged then recreated. This control information is recreated when you run **emfsconf**.

Figure 20-8**EpochMigration Local Configuration from the Disaster Report**

```

EpochMigration Local Configuration

EpochMigration System Configuration:
  Enable_stage_outMax_trailsEnable_self_describing
Y           3           N

Staging trail "Retrieve_random"
  Stage outs enabled: Y Media: EO      Unrestricted
  Self-Describing enabled: N
  Enable   HWM   LWM   PSWM   Delay   Mntpoint
Y       68    34    17      0 defaults for Retrieve_random

Staging trail "Retrieve_cached"
  Stage outs enabled: Y Media: EO      Unrestricted
  Self-Describing enabled: N
  Enable   HWM   LWM   PSWM   Delay   Mntpoint
Y       95    88    80      0 defaults for Retrieve_cached

Staging trail "Archive"
  Stage outs enabled: Y Media: EO      Unrestricted
  Self-Describing enabled: N
  Enable   HWM   LWM   PSWM   Delay   Mntpoint
Y       68    34    17      0 defaults for Archive

Staging trail "Trail_1"
  Stage outs enabled: Y Media: EO      Unrestricted
  Self-Describing enabled: N
  Enable   HWM   LWM   PSWM   Delay   Mntpoint
Y       94    78    64      0 defaults for Trail_1
Y       d     d     d       d /data
Y       95    88    80     20 /data2
Y       95    88    80     30 /data3
Y       90    68    45     70 /data7
Y       95    88    80     10 /data1
Y       95    88    80     60 /data6
Y       95    88    80     40 /data4
Y       95    88    80     50 /data5

```

6. Select a temporary location for LOCAL_DATABASE.

CAUTION: You cannot use a filesystem that is enabled for HSM for the temporary location of the LOCAL_DATABASE. (See Figure 20-8 on page 20-18 for the configuration of such a filesystem.)

CAUTION: The LOCAL_DATABASE backup must *not* be restored in place (to its original location), or the temporary database files that were just created and are being used to restore the original database are overwritten. This causes the disaster recovery process to fail.

You must have a filesystem with enough free disk space to restore the LOCAL_DATABASE backup to a temporary location. The amount of required disk space is noted in the LOCAL_DATABASE portion of the Disaster Report (see Figure 20-7 on page 20-16). If the filesystem you choose is the same filesystem that contains the EDM software, you need twice as much free disk space as noted in the Disaster Report.

7. Use **ebrestore** (*not* the EDM Restore window) to restore the LOCAL_DATABASE backup to the temporary location determined above, such as
/newpath/newdir/temp_local_db.

This restores critical files originally located in /usr/epoch.

```
emc# mkdir -p /newpath/newdir/temp_local_db
emc# ebrestore -w emc:LOCAL_DATABASE -c emc -D emc
-d /newpath/newdir/temp_local_db /
```

Note: In “-w emc:LOCAL_DATABASE,” “-c emc,” and “-D emc,” “emc” is the name of the server.

8. Now run a full label and barcode inventory to avoid duplicate entries in the catalog. You can run an inventory by using the **evminventory** command (refer to the **evminventory** man page), or through the Library Unit Manager window. (Refer to EDM online help for more information.)

9. Stop the following on the server as shown:

- a. Stop catalog processing by halting the **ebcatalogd** daemon. To do this, enter:

```
emc# ebcatalogd -halt
```

Note: **emfmd** must be running on HSM servers.

- b. List the applicable backup, volume management, and HSM daemons with the following command:

```
emc# edmproc
```

- c. Stop the daemons with the following command:

```
emc# edmproc -shutdown
```

- d. Verify that all processes are stopped.

```
emc# edmproc
```

On an HSM system, **emfmd** should still be running. On a system without HSM there should not be any daemons still running.

10. After the required shutdowns, move the LOCAL_DATABASE from the temporary location to its permanent location.

Note: There are two copies of **eb_disaster_move**. Be sure to use the copy relative to the temporary location of LOCAL_DATABASE as shown in these instructions.

CAUTION: Do NOT use the copy in:
/usr/epoch/EB/config/

To do this, change to the temporary location:

```
emc# cd /newpath/newdir/temp_local_db
```

Then find the path of the relative **eb_disaster_move**, cd to it and verify that you are there:

```
emc# find . -name eb_disaster_move -print  
emc# cd <path containing eb_disaster_move>  
emc# pwd
```

Move the LOCAL_DATABASE to its permanent location by using **eb_disaster_move**:

```
emc# ./eb_disaster_move
```

Note: Regarding use of duplicate volumes:

When `eb_disaster_move` completes, it automatically checks for failed and uncompleted duplications. If a duplication failed, you must use the original volume for restore. If the duplication had not completed or was scheduled, at the time the catalog was written to tape, the duplicate will be available for use.

If the disaster occurred soon after the backup completed, these duplications may not have completed. If there are errors reported during the restore while using the duplicate, use the original volume. If you do not have the original, contact Customer Service to restore from an earlier duplicate.

After you have completed the disaster recovery, reschedule the duplication of those original volumes that were not successfully duplicated.

`/newpath/newdir/temp_local_db` remains after **`eb_disaster_move`** is done but is no longer needed. You may wish to delete it after completing all disaster recovery procedures.

Any modifications to the `eb.cfg` file that were made after the LOCAL_DATABASE backup must be added manually to the restored `eb.cfg` file. You cannot restore any of the catalogs after the `local_db` backup (such as a new work item), until this is done.

Note: If you want to use duplicate volumes that were made after the LOCAL_DATABASE backup, follow the instructions in "Restore Catalogs and Backup Information Created After LOCAL_DATABASE Backup" on page 20-23.

11. Restart the following as shown:

- a. Restart the Volume Management daemon with the following command:

```
emc# sh /usr/epoch/etc/rcS/S20elm start
```

- b. Restart **ebfs** with the following command:

```
emc# sh /usr/epoch/etc/rcS/S30ebfs start
```

- c. If you have an HSM system, reenale staging with the following command:

```
emc# sh /usr/epoch/etc/rcS/S40mal start
```

You do not restart the **ebcatalogd** daemon at this time. It is restarted after you reboot the server. See “Stop Backups” on page 20-23.

12. Eliminate any incomplete backups or catalogs that could confuse catalog processing. To do this, enter:

```
emc# ebexpire -partial -purge -expire
```

13. Ensure that all catalogs that you just restored are in sync with the saveset database. To do this, enter:

```
emc# ebcatalogclean -disaster
```

This completes the restore of the LOCAL_DATABASE backup which contains the server catalogs and backup information at the time of the last LOCAL_DATABASE backup. The LOCAL_DATABASE is no longer available (or needed) for the restore process. If you need it again, start at step 6 on page 20-19.

At this point, the server's library units are configured as they were at the time of the last LOCAL_DATABASE backup. If there were no significant events after the LOCAL_DATABASE backup (Step C in Figure 20-1 on page 20-2), enter **ebcatalogd** on the command line to restart the daemon and skip to “Restore Data Created Before LOCAL_DATABASE Backup” on page 20-28.

Reconfigure Library Units

For changes after the last LOCAL_DATABASE backup

If any library units changed their configuration *after* the LOCAL_DATABASE backup was generated (step C in Figure 20-2 on page 20-3), use **lmconfig** to reconfigure them. There is no information in the Disaster Report to assist you in this step, since these changes (if they occurred) happened after the Disaster Report was generated.

Restore Catalogs and Backup Information Created After LOCAL_DATABASE Backup

For changes after the last LOCAL_DATABASE backup

If any of the events, such as D, E, F, or G in Figure 20-2 on page 20-3, occurred after your last LOCAL_DATABASE backup, you should get the catalogs and backup information back as soon as possible.

This applies, if you are using duplicate media which completed after the LOCAL_DATABASE backup.

To do this, you need to stop all backups, run a full inventory, import volumes, and then restore the catalogs and backup information.

Stop Backups

Disable backups by commenting out any **ebbackup**, **ebexpire**, and **ebcatclean** commands in the root crontab file. This needs to be done again because root crontab may have been modified when you moved LOCAL_DATABASE from the temporary location to the permanent location.

Reboot the server with the following command:

```
emc# /usr/sbin/shutdown -y -i6 -g0
```

This starts **vmdaemon** and **ebcatalogd** and makes all of your library units available to use in the remaining restore process.

Run a Full Inventory and Import Uncataloged Volumes

1. Take the box of tapes you set aside in step 2 on page 20-12 and insert the tapes in the library unit.

2. Inventory the contents of the library unit:

```
emc# evminventory -l at_dlt_452_0 -a -L
```

Where -l = library name, -a = all slots, and -L = Verify Label

3. Then import all uncataloged volumes in the library unit.

```
emc# evmimport -l at_dlt_452_0 -a
```

Where -l = library_name, -a = all slots (uncataloged only)

4. Wait until the inventories complete before proceeding.

Import Backup Catalogs and List Volumes

Next import the current catalogs for the current rotation.

1. To identify these, run **ebreport media** and select the most recent volume in the most recent rotation for each trail (as noted by a "*" in the example in Figure 20-9).

```
emc# ebreport media
```

Figure 20-9**Output from ebreport media**

EDM Backup Media Report for server missile on May 9 14:22:22 1999

Report options: none

Rotations for Template "usr_bin", Trail "usr_bin_DLT", Primary Trailset

09/30/1998 12:54:42 Rotation ID:4CD84987.F6BECF8D.00000200.54028F30, 4 backups

Media duplication used on 1 copy

*Orig Vol: 60D84A1170094B3E (BNY574), Seq #: 000024 in TLU: at_dlt_3264_0, media: DLT

Dup Vol: 73D8745B3E0384A5 (BDE133), Seq #: 000028 in TLU: at_dlt_3264_0, media: DLT

Duplication State: Done, Successful, Duplication Date 05/08/1999 16:06:04

Orig Vol: 4CD84987F6BECF8D (ASV891), Seq #: 000027 in TLU: at_dlt_452_0, media: DLT

Dup Vol: 96D8746209A96A98 (BDE128), Seq #: 000031 in TLU: at_dlt_452_0, media: DLT

Duplication State: Done, Successful, Duplication Date 05/08/1999 16:25:04

.
.
.

EDM Baseline Media Report for server adam on May 9 13:10:26 1999

2. Run **ebimport** to import the backup catalogs and information from the volumes that were current at the time of the last successful LOCAL_DATABASE backup (identified by an * in the output from **ebreport media**, as shown in the above example).

These volumes may also contain appended backups not known at the time of the last LOCAL_DATABASE backup. (These volumes may reside inside or outside the library units.)

Note: This portion of the restore can take a considerable amount of time because EDM Backup has no record of the backups or the volumes on which they reside.

3. For each volume, run the following command to import catalogs and backup information created after the time of the LOCAL_DATABASE backup:

```
emc# ebimport -media valid -clever 9 -level 9
```

Some catalog imports may fail. If so, stop the catalog daemon and restart it with the **30 second** option, which causes these catalogs to be processed again in 30 seconds rather than the default value of one day.

```
emc# ebcatalogd -halt
```

```
emc# ebcatalogd -retry_time 30
```

If you make this change you should reset **ebcatalogd** after completion of the disaster recovery. To do this, enter:

```
emc# ebcatalogd -halt
```

```
emc# ebcatalogd -catalogs 3 >& /dev/null
```

4. Execute the following command to list all volumes in the library unit and their states:

```
emc# evmstat -v
```

If there are any “uncataloged” volumes that are allocated after the last LOCAL_DATABASE backup, you must restore the catalogs and backup information. Otherwise skip to “Restore Data Created Before LOCAL_DATABASE Backup” on page 20-28.

Restore Catalogs and Backup Information

Restore all catalogs and backup information that were created or changed *and* backed up, staged, or part of a baseline backup after the LOCAL_DATABASE backup completed. (See Figure 20-2 on page 20-3.) This is restoring data from volumes that were allocated after the last LOCAL_DATABASE backup.

Note: This portion of the restore can also take a considerable amount of time.

1. Run a **dbreport volume**, sort the output and redirect to a file as follows:

```
emc# dbreport volume | sort > pre.sort
```

2. Use the EDM Library Unit Manager window in the EDM GUI to import all uncataloged volumes, when a backup was appended to a previous backup after the LOCAL_DATABASE was backed up.

Note: If you are doing a partial disaster recovery, and are skipping over portions of this chapter, be sure that the value of VM_ALLOW_DUP_SEQ_IMPORT in /usr/epoch/etc/vm/vm.cfg is set to "no," which is the default setting that allows the overwriting of a duplicate sequence number.

Look at which volumes are offline. Select all uncatalogued volumes and import them. When a message appears asking if you want to overwrite a duplicate sequence number, answer **yes**. This makes the latest backup to that trail available.

3. Run another **dbreport volume**, this time redirecting the sorted output to post.sort:

```
emc# dbreport volume | sort > post.sort
```

4. Run a **diff** on the two files:

```
emc# diff pre.sort post.sort
```

Examine the output for volumes that changed from "available" to "allocated." The following is an example of **diff** output that shows an EB volume that changed from "available" to "allocated:"

media	application	volume_name	seq	side	barcode	state	valid
DLT	EB	Server_alt_DLT	147	0	00000327	available	2BC53BEA44E275B4
DLT	EB	Server_alt_DLT	147	0	00000327	allocated	123456EA44E275B4

The second column indicates which EDM application uses the volume. EB indicates EDM Backup, EM indicates an HSM volume, and baseline indicates a baseline backup.

For EDM Backup volumes that changed, enter:

```
emc# ebimport -media valid -clevel 9 -level 9
```

For EM or baseline volumes that changed, enter:

```
emc# ebfs_import -v valid
```

Run **ebcatalogd -status** until it reports that all catalogs were processed.

At this point, you have restored all of your most recent server catalogs and backup information that were complete at the time of the last LOCAL_DATABASE backup and those completed after the last LOCAL_DATABASE backup.

Restore Data Created Before LOCAL_DATABASE Backup

You need to restore user files and some catalogs which were created before the LOCAL_DATABASE backup.

Before you begin this section, restart the ebcatalogd daemon if you did not do so in the previous section. To restart the daemon, enter the following command at the command line:

```
# /usr/epoch/EB/config/daemon_startup
```

1. Use **ebrestore** with the overwrite option set to never to restore each server work item you need to restore your filesystems and partitions, and any customized operating system files such as /.login, /.cshrc, /.profile, network databases, and crontab files as they were at the time of the last LOCAL_DATABASE backup.

```
# ebrestore -o never -i
```

The **-o never** option ensures that you do not overwrite any files that you already restored, such as / and /usr which were reinstalled in Step 2 on page 20-6. Overwriting them may crash the system.

CAUTION: You use the -o never option so that you do not restore all of / and /usr; since doing so may crash the system. For example, if you overwrite /usr/lib/libc.so.1 on a Solaris machine you crash the system.

However, you may want to restore some specific individual files selectively with custom settings (such as /etc/hosts and

**/etc/passwd) or programs loaded into
/usr/local or /usr/etc for local use using
ebrestore with overwrite set to always.**

At this point, you should have restored all filesystems and partitions that were present on the server up to the last LOCAL_DATABASE backup. (If you have an HSM system, EM and HSM now reattach properly.)

2. If an entire HSM VxFS filesystem was lost, you can do the following to restore their data quickly:

- a. Remount the filesystem without logging:

```
emc# mount -F vxfs -o remount,nolog /home1
```

- b. Perform the restore using **ebrestore** or the EDM Restore window.

- c. Remount the filesystem with logging when the restore is complete:

```
emc# mount -F vxfs -o remount,log /home1
```

3. If you restored any system files, reboot the server.

Note: Do not attempt multiple restores from the same trail simultaneously. This slows down the restores significantly.

Reenable crontab Entries

Reenable backups by undoing any edits you made to the root crontab file in “Disable Activity” on page 20-5 or “Stop Backups” on page 20-23.

If you do not want to restore past catalogs that allow you to restore older data, the recovery of the server is complete at this point.

Restore Past Catalogs

If you want to restore your earlier catalogs (which then allows you to restore earlier backup data), continue with this section. These backup catalogs provide a history of backups completed prior to the last LOCAL_DATABASE backup.

You know which catalogs were not yet restored because **ebreport history** may report savesets with ??????? in the Entries field. This indicates that the catalogs for those backups must be imported or restored before the backups can be restored.

1. Use **ebrestore** to restore the rest of the backup catalogs. Set the overwrite option (-o) to never.

```
emc# ebrestore -c emc -D emc -o never -d / -w emc:/data
/data/epoch/EB/catalogs /
```

In this case, emc:data is the name of work item that backed up the partition on which the EDM Backup catalogs were stored, and /emc/data/epoch/EB/ is where the EDM Backup catalogs were stored (see the example in Figure 20-5 on page 20-9).

At this point you restored all of the backup catalogs up to the last LOCAL_DATABASE backup.

2. Run **ebexpire -partial -purge -expire** to eliminate any incomplete backups or catalogs that could confuse catalog processing.
3. Make sure all catalogs were processed.

- a. Run **ebcatproc -any** to start catalog processing for these particular savesets.
- b. Run **ebcatalogd -status**. Continue when **ebcatalogd** reports that all catalogs were processed.

ebcatalogd may indicate that some catalogs failed. If so, run **ebcatalogd -halt** to stop the catalog daemon and restart it with the **-retry_time 30** option, which causes these catalogs to be processed again in 30 seconds rather than the default value of one day.

If you make this change you should reset **ebcatalogd** after completion of the disaster recovery. To do this, enter:

```
emc# ebcatalogd -halt
emc# /usr/epoch/EB/config/daemon_startup -ebcatalogd
emc#
```

At this point, you restored the full EDM Backup catalog system (except for the most recent remote client catalogs which are discussed next).

4. Now, you can restore any backed-up data that was *not* present on the server at the time of the LOCAL_DATABASE backup, by using the **ebrestore -o never** command.
- ```
emc# ebrestore -o never
```

---

## Restore Missing Catalogs

As a final step to restoring your data in a disaster situation you must make sure that you restored all of the missing backup catalogs. These backup catalogs were neither backed up by the work item for the partition containing the catalogs nor were they part of the LOCAL\_DATABASE backup.

1. To determine which catalogs were not backed up prior to the LOCAL\_DATABASE backup, run the **ebreport history** command with the **-ebimport** option.

```
emc# ebreport history -ebimport
```

This report displays ?????? in the Entries field if the saveset indicates the backup was complete but the catalog has not yet been restored.

In the example shown in Figure 20-10 on page 20-33, nine backups are missing their catalog files.

2. Identify the savesets that were not restored and import each one by using the following command:

```
emc# ebimport -clever 9 -level 9 -ok_if_unexpired saveset_id_1 ...
saveset_id_n
```

This restores the catalogs even if the saveset database indicates they are not yet expired.

Figure 20-10

## Output from ebreport history -ebimport

```

**** Work Items for Template Epoch_site, Primary Trailset ****
**Item "dilbert"

Time Lvl ID Status Entries Expires serverdb
1/ 3/98 20:56 0 5542FA53.2D28D220 sorted ??????? 1/ 3/99
SaveSet ID 5542FA53.2D28D220

**Item "lamborghini"
Time Lvl ID Status Entries Expires serverdb
7/ 3/97 0:38 0 5542FA53.2C350E95 complete ?????? 7/ 3/99
5/31/97 19:20 0 5542FA53.2C0A93DB complete ?????? 5/31/99

**Item "odie"
Time Lvl ID Status Entries Expires serverdb
6/14/97 21:31 0 5542FA53.2C1D26F6 complete ?????? 6/14/90
5/17/97 19:40 0 5542FA53.2BF82217 complete ?????? 5/17/90

**Item "support"
Time Lvl ID Status Entries Expires serverdb
7/16/97 23:31 0 5542FA53.2C4773D6 complete ?????? 7/16/99
6/17/97 21:44 0 5542FA53.2C211F44 complete ?????? 6/17/99
5/20/97 19:42 0 5542FA53.2BFC1897 complete ?????? 5/20/99
4/30/97 19:49 0 5542FA53.2BE1BBA0 complete ?????? 4/30/99

Of the backups listed, 9 are missing catalog files.

Those backups whose count of catalog entries are filled with "?"
cannot be recovered until their catalogs have been recovered by running
"ebimport -ok_if_unexpired" on the backup's saveSet ID.

```

For information on restoring a UNIX client see Chapter 21  
 "Recovering a UNIX Client from Disk Failure."



---

## 21 Recovering a UNIX Client from Disk Failure

This chapter contains disaster recovery procedures to use when you experience a disk crash on a UNIX client. Because each disaster is unique, these steps are presented only as a guide and not as all-inclusive, step-by-step instructions.

**CAUTION:** Performing a disaster recovery requires experience with EDM Backup administration (and HSM administration, if you have the HSM option), UNIX system administration, and the site environment.

In general, to restore a lost EDM client, you must replace the damaged disk, reinstall the operating system, reinstall (or relink) to any lost EDM client software, and use the EDM Restore window to restore your lost files.

If you also need to restore the backup server, be sure to restore the server first, *before* you restore the client. The following procedures assume that the EDM Backup server is not damaged or was already restored.

**Note:** This chapter applies to UNIX clients. Refer to the appropriate client supplements for recovering other clients.

---

## Recovering a Client

Because you perform all of the restore in multiuser mode, you must notify the user community to stop all activity on the client. No one should be able to log in. The client must be inactive before you perform any disaster recovery procedures.

The following procedures also assume that all of the client filesystems are damaged or unavailable. If some filesystems remain intact, you might be able to skip some steps. These are guidelines only.

---

### Beginning Steps

To restore an EDM client in case of a disk crash or major loss of files:

1. Edit root's crontab file on the server to comment out all backup and staging commands that might start up automatically. Comment out the following entries (note that some are HSM-specific (begin with em) and apply only to EDM Migration clients):

```
00 0 * * * /bin/kill -1 `cat /usr/epoch/etc/mal/emmasterd.pid` >/dev/null
2>&1#EPCmalc

15 23 * * * /usr/epoch/bin/emvck >/dev/null 2>&1#EPCmalc

00 1 * * * /usr/epoch/bin/emcompact -c >/dev/null 2>&1#EPCmalc

00 2 * * * /usr/epoch/bin/emscheck >/dev/null 2>&1#EPCmalc

00 3 * * * /usr/epoch/bin/emsundel >/dev/null 2>&1#EPCmalc

00 18 * * * /usr/epoch/EB/bin/ebbackup default >/dev/null 2>&1 #EPCebs

00 11 * * * /usr/epoch/EB/bin/ebexpire -expire -purge >/dev/null 2>&1 #EPCebs

00 1 * * * /usr/epoch/EB/bin/ebcatclean -fix_saveset >/dev/null 2>&1 #EPCebs

00 3 * * * /usr/epoch/EB/config/local_db_warning >/dev/null 2>&1 #EPCebs
```

2. Check hardware and replace if necessary. Run a disk diagnostics program. Be sure to identify and replace all damaged hardware *before* performing any software recovery procedures. Make sure replacement hardware is fully compatible with the system you had before the disaster. Each new disk should have at least the same storage capacity as the old disk.
3. Reinstall the client's native operating system. For directions, see the documentation that is supplied with the client. This reinstall process partitions your disk, restores the root filesystem on the client, and restores that portion of */usr* that the operating system loads. Make sure to specify that these filesystems are at least as large as and have the same (or larger) number of inodes as they had before the disaster.

**CAUTION: Do not continue with software recovery until you are absolutely certain the disk is free of hardware problems.**

If you are recovering a Backup client, go to step 6.

If you are recovering an HSM client, go to step 4.

---

### **For HSM Clients Only**

4. If the client is an EDM Migration client, reinstall VERITAS VxFS filesystems. Customer Service has the *Software Installation* manual, call them with questions.

Go to step 6.



## For Backup and HSM Clients

5. Reinstall the EDM Backup software using **eb\_install\_client**. Reconfigure EDM Backup on the client, using exactly the same settings as those defined prior to the disaster.

Refer to the full **ebreport disaster** report as necessary, for a list of the workitems prior to the disaster.

**CAUTION: Do *not* restore all of / and /usr; doing so may crash the system. Only restore those files that contain customized settings; for example, /etc/hosts and /etc/passwd, or programs that are loaded into /usr/local or /usr/etc for local use.**

6. Create and mount new, empty filesystems that are identical to (in terms of number of blocks and inodes), or larger than, those on the client before the disaster. Note that the commands used and command syntax vary from platform to platform. For a Sun workstation, for example, use **newfs**, **mkdir**, and **mount**. Refer to the **ebreport installation** or a full **ebreport disaster** report for a list of client filesystems and their sizes prior to the disaster.
7. If the filesystem that contained the backup software is still intact, but /usr was destroyed, you need to recreate the symbolic link from /usr pointing to this filesystem that contains the backup software.

If you are recovering a Backup client, go directly to step 12.

If you are recovering an HSM client, go to the next section, "For HSM Clients".

---

## For HSM Clients

When recovering an HSM client, verify whether the EDM software (/ep\_usr) is damaged.

- If it is not damaged, use the following steps in this order: step 9., on page 21-5 step 10., on page 21-5 and step 5., on page 21-4
- If it is damaged, use the following steps in this order: step 8., on page 21-5 step 10., on page 21-5 step 5., on page 21-4 step 11., on page 21-5 and then step 9., on page 21-5

8. Reinstall the EDM Migration (mc) software using **ep\_install**.

9. Run **emlsconf** to display the old configuration. Any newly-created filesystems must be reconfigured for migration even if the MAL database files look intact. The commands are:

```
emlsconf -r filesystem
emfsconf filesystem old_parameters
```

where *old\_parameters* are the **emfsconf** parameters displayed by the **emlsconf** command.

10. Shut down the HSM daemons prior to restoring the EDM software filesystem:

```
emc# sh .usr/epoch/etc/rcK/K60mal stop
```

11. Restore the filesystem where the EDM software was installed (/usr/epoch/etc) out of place, and then move it to the proper location (/usr/epoch) to bring back the database files. Then reboot the migration client.

12. If you have an EDM database client, reinstall the vendor database software, if needed, via their instructions. Then reinstall the Backup software, if needed. For offline database backup and Oracle online backup, you can do this through the EDM Backup Configuration window. For earlier online backup clients, see the appropriate online backup supplement and release notes.

---

## For Both Backup and HSM Clients

After you reinstall the EDM Backup software using **eb\_install\_client**, you are ready actually to restore your files.

13. Open the EDM Restore window:

- from the client using:

```
client1# edmcrestore
```

or

- from the EDM Main window.

Use the EDM Restore window or **ebcrecover** to restore your lost filesystems and files. In the EDM Restore window you can browse a list of work items for the client, select the work item, mark the files you want to restore, and start the restore.

14. Reboot the client system.

---

---

## **Part VI**

# **Appendixes**

---

---

# **A** **Directory Structure**

This appendix describes the directory structure of EMC Data Manager backup, volume management, and HSM.

This is the directory structure of the software that runs on a server and the software that runs on the client, which enables the client to collect backup data and restore recovered files.

This appendix describes the following:

- Backup Server Directory Structure
- Backup Client Directory Structure
- Volume Management Directory Structure
- HSM Directory Structure

---

## Backup Server Directory Structure

The EDM backup server software handles all aspects of backup and restore operations. Its components are:

- EDM transfer protocol or remote shell (**rsh**) – Communication between the server and client. The EDM transfer protocol is supported on UNIX platforms.
- EDM Backup server processes – Daemons and programs that service backup and restore operations.
- The Global Configuration Database – Configuration specifications that direct backup and restore operations and files that manage the database.
- The Installation Directory – Directory on the EDM server that contains backup installation and executable files for both the server and the client systems. This section describes backup by looking at the elements on the server that operate the backup and restore operations.

EDM stores two sets of files on the server: large and small. The set of large files (backup catalogs and log files) grows over time and the set of small files (client binaries and configuration files) does not.

---

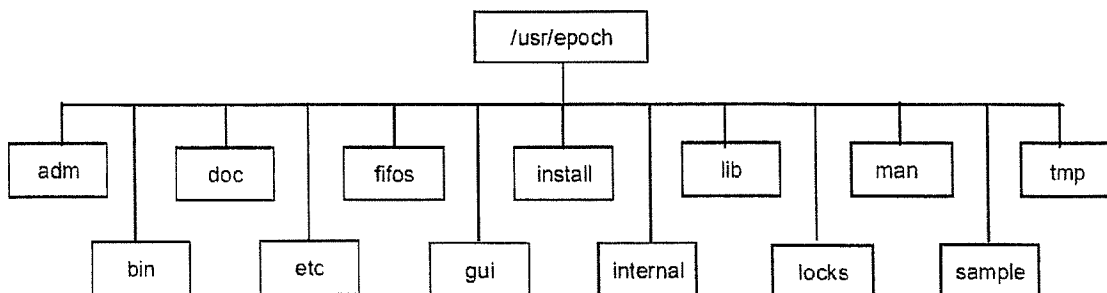
### **/usr/epoch**

The /usr/epoch directory includes several subdirectories that contain configuration and database files.

**CAUTION: Although these files are text files, you should never attempt to modify them with an ordinary editor. The configuration commands and the EDM Backup Configuration window do more than just modify the files.**

Figure A-1 shows many of the top-level directories.

Figure A-1

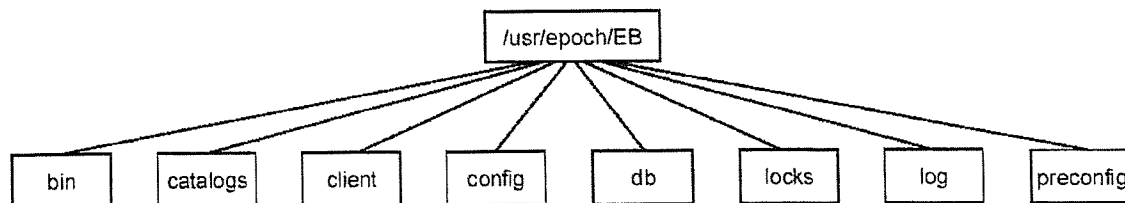
**Server Directory Structure (/usr/epoch)**

- adm – location for circular and archived log files
- bin – contains user executable binaries
- doc – contains online documentation
- etc – contains vital EDM software databases
- fifos – EDM software fifos
- gui – gui resource files
- install – contains non user executable install files
- internal – contains EDM internal commands
- lib – contains shared libraries and non user executable daemons (linked to from /usr/epoch/etc/lm)
- locks – contains lock files
- man – contains all man pages
- sample – contains sample Volume Manager and Library Manager configuration files
- tmp – EDM software tmp directory

**/usr/epoch/EB**

The next level of backup server directories, /usr/epoch/EB, contains the following subdirectories.

---

**Figure A-2****Server Directory Structure (/usr/epoch/EB)**

- bin – contains the server executables
- catalogs – contains the backup catalogs and the saveset database (ebsaveset\_db)
- client – contains client executables and installation files
- config – contains the configuration files, including eb.cfg
- db – contains database files; for example, cattask.list and htab
- locks – contains lock files; for example, ebcatalogd.lock
- log – contains the backup log files
- preconfig – contains a file for each client with default configuration information

When you initially configure backup using **eb\_server\_config**, you can create the directory /usr/epoch/GENDIR/EB to hold large catalogs and log files.



If you run **eb\_server\_config** without the **-D** option, you are prompted to create the alternate directories. **eb\_server\_config** also creates the following symbolic links (symlinks):

- /usr/epoch/EB/catalogs is a symlink to /usr/epoch/GENDIR/EB/catalogs
- /usr/epoch/EB/log is a symlink to /usr/epoch/GENDIR/EB/log

Refer to the **eb\_server\_config** man page for more information.

---

## Bin Directory

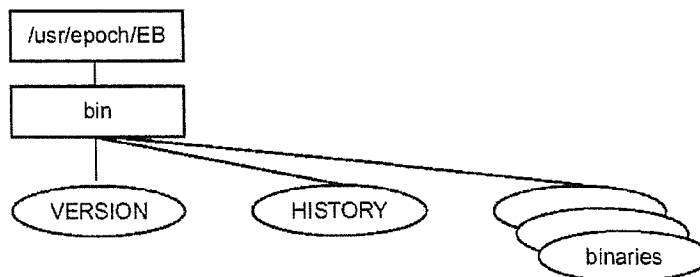
The bin directory contains all the server binaries and a VERSION file that contains the current version of EDM Backup software and a HISTORY file that contains a history of all EDM Backup installations.

---

Figure A-3

---

The bin Directory



---

## Catalogs Directory

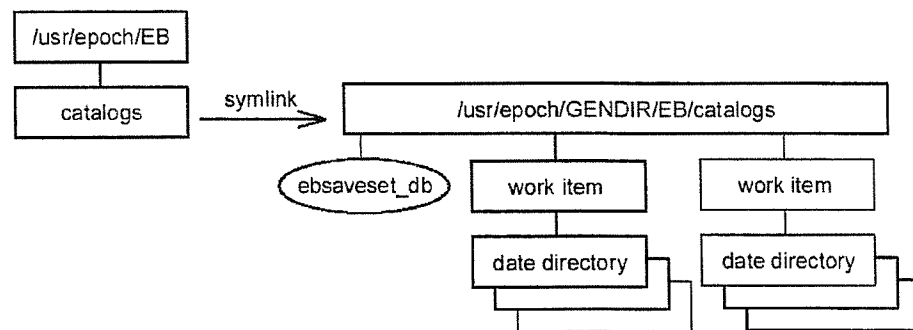
The catalogs directory contains the saveset database and one directory for each configured work item. This subdirectory has the same name as the work items and is created as necessary by the catalog subsystem. Any "/" characters in the work item's name are translated to "%" characters for use in directory and filenames.

Below the work item directory are DATE directories. DATE directories are named with four-year digits followed by a dash and the month digits (for example, 1998-02). The catalog software must be able to create a date directory when it writes a new backup catalog.

The purpose of this directory structure is to enable system administrators and customer support personnel to scan directories of moderate size for catalog files, while allowing the software a simple and fast algorithm for locating catalogs.

Figure A-4

The catalogs Directory



## Client Directory

The client directory contains binaries and other client platform-specific software. Under this directory are client platform directories of the form:

*manufacturer\_cpu\_os[\_other]*

The `_other` suffix is optional. It is used for anything that does not fit into the scheme of manufacturer, cpu, and OS. Normally, it is not used. Client platform directories have names such as `sun_sun4_v5` or `hp_800_v10`.

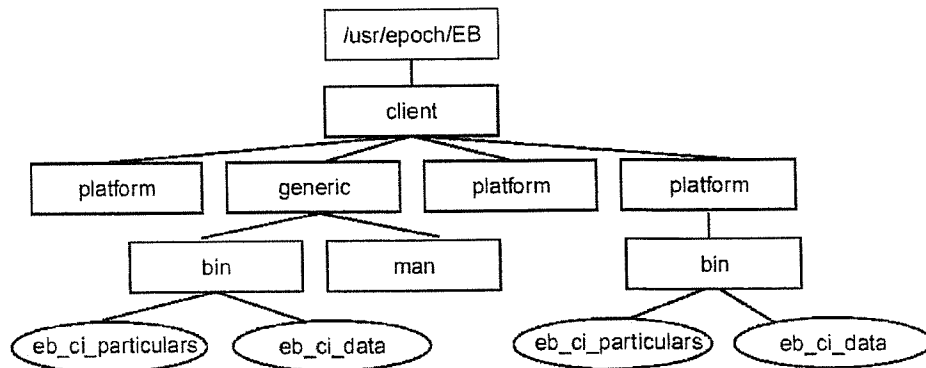
The client directory also contains a subdirectory called generic. This subdirectory (as well as each of the platform subdirectories) contains a bin subdirectory, which contains up to two files that are used in customizing the client installation process. The file that is shipped as part of the backup software distribution package is named eb\_ci\_particulars. The second file is generated from the first and is named eb\_ci\_data. The following describes each file:

- eb\_ci\_particulars file – Contains definitions set by EDM for constants that should not have to be changed at your site (such as the name of the platform-specific options to be used with the **mmtpts** command).
- eb\_ci\_data file – Contains the generic platform's constants. This file is created during backup installation for each platform type under /usr/epoch/EB/client.

Under each specific client platform directory are the subdirectory, bin and optionally, the subdirectory man. The bin directory contains all client scripts and binaries for installation of the client as well as those for the normal backup operation of the client. Like the bin subdirectory under generic, these bin directories may also contain the eb\_ci\_particulars file. In all cases this subdirectory contains the eb\_ci\_data file. The man directory contains man pages for the client recovery program(s).

Figure A-5

The client Directory

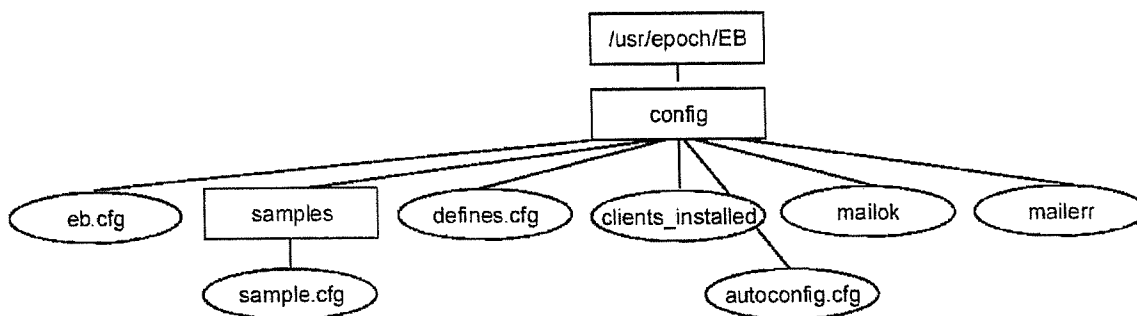


## Config Directory

The config directory contains the default `eb.cfg` configuration file, sample configuration files, and a sample **findxcpio** macro file, called `defines.cfg`. It also contains a file called `clients_installed` which is modified when each client is installed. It includes the **mailok** script to which the backup software passes backup completion reports, and the **mailerr** script to which it passes backup failure reports.

Figure A-6

The config Directory



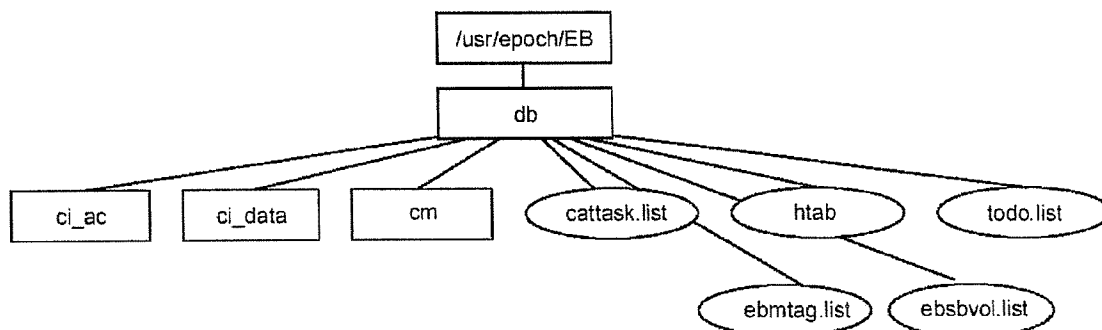
If you installed backup to backup all (or some) clients by using the autoconfig facility, this directory also contains a default configuration file for each such client (`/config/client-name/dcfg`), and an `autoconfig.cfg` file. This `autoconfig.cfg` file references the default (`dcfg`) file for each client, via an `#INCLUDE`, and provides a single work-group definition for the `auto_configured_work_group`. You can then create a template that references that work item and thus backs up all the autoconfigured clients at once.

## Db Directory

The db directory contains the EDM backup database files.

Figure A-7

The db Directory



- `ci_ac` is a subdirectory that contains information that is used by `recover`'s graphical user interface. It stores information similar to that in the `preconfig` directory but is more tuned toward graphical standards.
- `ci_data` is a subdirectory that stores the installation parameters for each client machine. These parameters describe what occurred during the most recent client installation. One file in this subdirectory exists for each machine installed, and it is named after the machine:  
`/usr/epoch/EB/config/db/ci_data/client-name`.

- `cm` is a subdirectory that contains all of the information that the **coverage** report uses.
- `cattask.list` is **ebcatalogd**'s database file. It contains all of the information that is needed to process the catalogs that **ebbackup** generates.
- `htab` lists and identifies all of the installed clients.
- `todo.list` is **ebbackup**'s database file. It contains all of the backup history, including when the backups occurred, how long they took, the old and current schedules, etc.

The following files exist only on an HSM system:

- `ebsbvol.list` (the saveset-to-baseline relations file) is used for baseline backups (levels B1 and B2). For each physical volume that is used in a baseline backup, this database records the volume ID and saveset ID of the backup.
- `ebmtag.list` maps work item names to EpochMigration tags.

---

## Locks Directory

The locks directory contains lock files, for example, `ebcatalogd.lock` and a lock file for each work item. This directory is flat; no subdirectories exist under locks.

---

## Log Directory

The log directory (which may contain a symbolic link to `/usr/epoch/GENDIR/EB/logs`) contains all of the EDM backup logs files. For more information on log files, see page 16-35.

---

## Preconfig Directory

The preconfig directory contains a file for each client. This file contains default configuration information for the client.

---

## Server Man Directory

The server man pages are installed in `/usr/epoch/man`.

For a listing of both server and client man pages, see Chapter 18 "Man Page Listing".

## Table of Backup Server Directories and Files

Table A-1 describes each of the server's configuration files.

**Table A-1**

### The Server's Configuration Directory

| File or Directory Name                             | Description                                                                                                                                                                                                                                               |
|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /usr/epoch/EB                                      | Contains the server's EB directories and files.                                                                                                                                                                                                           |
| /usr/epoch/EB/bin                                  | Contains the server's binaries.                                                                                                                                                                                                                           |
| /usr/epoch/EB/bin/VERSION                          | Contains the current installed EDM backup version.                                                                                                                                                                                                        |
| /usr/epoch/EB/bin/HISTORY                          | Contains the EDM backup installation history.                                                                                                                                                                                                             |
| /usr/epoch/EB/catalogs                             | May contain a symbolic link to the backup catalog directory on the EDM backup server: /usr/epoch/GENDIR/EB/catalogs<br><br>This stores the backup catalogs.                                                                                               |
| /usr/epoch/EB/client                               | Contains all client installation files. Contains a subdirectory for each platform manufacturer, CPU architecture, and OS revision supported for EDM backup clients.                                                                                       |
| /usr/epoch/EB<br>/client/ <i>manuf_cpu_os</i>      | Describes the platform manufacturer, CPU architecture, and OS revision. Each subdirectory name is of the format: <i>manufacturer_cpu_os</i> (e.g., <i>sun_sun4_v5</i> ). This directory is not used until the client installation procedure is performed. |
| /usr/epoch/EB<br>/client/ <i>manuf_cpu_os</i> /bin | Contains platform manufacturer client binaries.                                                                                                                                                                                                           |
| /usr/epoch/EB<br>/client/ <i>manuf_cpu_os</i> /man | Contains the client man pages.                                                                                                                                                                                                                            |
| /usr/epoch/EB/client/generic/bin                   | Contains binaries that are used to customize client installations.                                                                                                                                                                                        |
| /usr/epoch/EB/config                               | Contains the EDM backup configuration files.                                                                                                                                                                                                              |
| /usr/epoch/EB<br>/config/autoconfig.cfg            | Contains information used by autoconfig to define the <i>auto_configured_work_group</i> , and to reference the default configuration file for each autoconfig client.                                                                                     |
| /usr/epoch/EB<br>/config/ <i>client-name</i> /dcfg | Describes the configuration for each client that is backed up via the autoconfig facility.                                                                                                                                                                |

---

**Table A-1**                      **The Server's Configuration Directory (Continued)**


---

| File or Directory Name                     | Description                                                                                                                                                 |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /usr/epoch/EB/<br>config/clients_installed | Contains an entry for each EDM backup client installed. You can read but not edit this file.                                                                |
| /usr/epoch/EB/config/defines.cfg           | Contains a sample findxcpio macro file.                                                                                                                     |
| /usr/epoch/EB/config/mailerr               | Contains the script to which the backup software passes backup failure reports.                                                                             |
| /usr/epoch/EB/config/mailok                | Contains the script to which the backup software passes backup completion reports.                                                                          |
| /usr/epoch/EB/config/samples               | Contains sample configuration files to use as a reference when editing the eb.cfg file.                                                                     |
| /usr/epoch/EB/config/eb.cfg                | Contains the server configuration file. You edit this file by way of the EDM Backup Configuration window to specify the server's configuration information. |
| /usr/epoch/EB<br>/config/local_db_startup  | Dictates what gets backed up by the LOCAL_DATABASE backup.                                                                                                  |
| /usr/epoch/EB<br>/config/local_db_cleanup  | Dictates what happens after the LOCAL_DATABASE backup is done.                                                                                              |
| /usr/epoch/EB<br>/config/local_db_warning  | Mails a warning message to the administrators if the LOCAL_DATABASE backup is too old.                                                                      |
| /usr/epoch/EB/db                           | Contains <b>ebbackup</b> 's database files.                                                                                                                 |
| /usr/epoch/EB/db/ci_ac                     | Contains information used by the EDM Restore window.                                                                                                        |
| /usr/epoch/EB/db/ci_data                   | Stores the installation parameters describing what took place during the most recent installation of each client machine.                                   |
| /usr/epoch/EB/db/cm                        | Contains all the information used by the <b>coverage</b> report.                                                                                            |
| /usr/epoch/EB/db/ebmtag.list               | Maps work item names to EDM Migration tags.                                                                                                                 |
| /usr/epoch/EB/db/todo.list                 | Contains all the backup history (when backups occurred, how long they took, schedules, etc.).                                                               |
| /usr/epoch/EB/db/ebsbvol.list              | Stores the volume id(s) and saveset ID for each baseline backup.                                                                                            |

---



Table A-1

The Server's Configuration Directory (Continued)

| File or Directory Name        | Description                                                                                                                                                     |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /usr/epoch/EB/db/cattask.list | Contains all information needed to process the catalogs generated by the backup software.                                                                       |
| /usr/epoch/EB/locks           | Contains all of the temporary lock files that the backup software creates during backup and restore operations. This includes locks on templates and trailsets. |
| /usr/epoch/EB/log             | May contain a symlink to the log file directory on the EDM backup server: /usr/epoch/GENDIR/EB/logs<br><br>Stores the server log files.                         |
| /usr/epoch/EB/preconfig       | Contains a file for each client; each file contains default configuration information for the client.                                                           |
| /usr/epoch/man                | Contains the man and cat directories for the man pages.                                                                                                         |

## Backup Client Directory Structure

All backup files on the backup client are stored in the backup client's home directory.

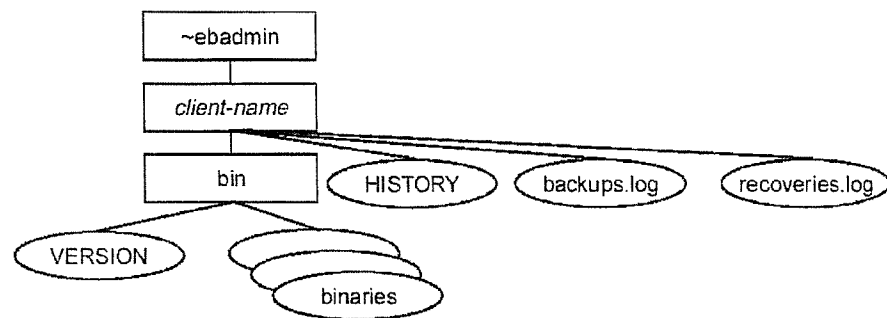
The examples in this document specify the name of the default user, ebadmin. The home directory is specified as ~ebadmin.

**Note:** For security purposes, ebadmin should not have a password, and a user should not be able to log in. The backup software creates the default user name with no password. (Refer to "Security Issues" below.)

## Client Home Directory

The goal of the client installation procedure is to set up the backup client home directory as shown below. Assuming that you used the default backup client username ebadmin, the home directory is ~ebadmin. You can have a separate backup client home directory on each client system or you can specify

one directory that all the clients mount via NFS. The directory structure allows this type of sharing, although it does not allow one client to be backed up by multiple EDM units.

**Figure A-8****The ~ebadmin Directory**

Under the home directory there is a subdirectory for each client, named with the client's network host name. The client directory contains the bin directory which holds all the client binaries as well as a VERSION directory that displays the versions of the binaries. The client directory also contains the backups and recoveries log files (for more information, see "Log Files" on page 16-35) and a HISTORY file that displays each client installation.

In the case where many clients share the same platform and OS release, installing binaries in each client's home directory is redundant. Thus, you can replace a bin directory with a symbolic link to a bin directory of another client of the same type. You must do this manually (it is not done during installation).

If you reinstall or deinstall a client, the existing bin directory is removed. Thus, any clients that have links to that directory are not backed up.

---

## Security Issues

A potential security hole exists for backup client directories that are shared via NFS by more than one client. For example, consider two clients, client1 and client2, that share the same backup client home directory. Call the backup client user ebadmin and this home directory ~ebadmin. In this case, someone who is logged in as root on client1 can go into the ~ebadmin directory on client1 and change its .rhosts file to allow access to the directory as ebadmin from client1 (normally access is only allowed from the EDM). He or she can then contact client2 as ebadmin from client1 and cause client2 to perform backups of any file on client2 and send the output to client1. Furthermore, the backup data can then be manipulated on client1 and sent back to client2 as a restore so as to overwrite existing files on client2.

**Note:** If this type of security is of concern to you, do not share client backup home directories via NFS.

---

## Additional Client Software

The default location for ~ebadmin is /usr/epoch/EB/CLIENT\_HOME.

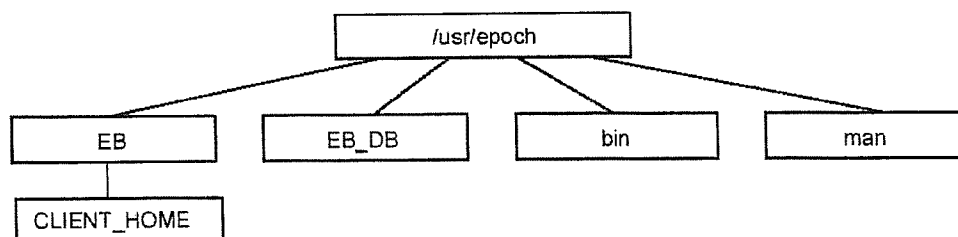
Some of the other client directories are listed in Figure A-9.

---

Figure A-9

---

Client Directory Structure



- EB – contains the client home directory (~ebadmin)
- EB\_DB – holds database backups
- bin – contains report and restore scripts
- man – contains client man pages

---

**Client EB\_DB Directory**

Each installed client gets the /usr/epoch/EB\_DB directory which holds database backups.

---

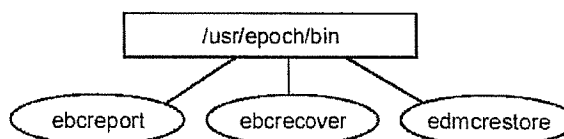
**Client bin Directory**

You can also install scripts on selected clients for client file self-recovery and report generation. By default, the scripts are installed in /usr/epoch/bin. You can also install the scripts in an alternate directory.

---

**Figure A-10**

---

**The Client bin Directory**

---

**Client man Directory**

You can choose to install the client man pages in either /usr/epoch/man, /usr/local/man, or in a location of your choice.

For a listing of both server and client man pages, see Chapter 18 "Man Page Listing".

## Table of Backup Client Directories and Files

Table A-2 describes these client files and directories.

**Table A-2**

**Client Directories**

| File or Directory Name                                       | Description                                                                                                                                                                                                                                                 |
|--------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ~ebadmin (/usr/epoch/EB/CLIENT_HOME)                         | Default client home directory that contains a subdirectory for each client, named with the client network host name.                                                                                                                                        |
| ~ebadmin/ <i>client-name</i> /bin                            | All client binaries for the client are stored here together with a VERSION directory showing the version of these binaries.<br><br>The bin subdirectory can be a symbolic link to the bin subdirectory of another client with the same architecture and OS. |
| ~ebadmin/ <i>client-name</i> /HISTORY                        | HISTORY file shows each installation that is performed for the client.                                                                                                                                                                                      |
| ~ebadmin/ <i>client-name</i> /backups.log                    | Client's backups.log file records critical information about backups performed on this client.                                                                                                                                                              |
| ~ebadmin/ <i>client-name</i> /backups.log.cminfo             | Contains coverage monitor data that is sent to the server after each backup.                                                                                                                                                                                |
| ~ebadmin/ <i>client-name</i> /recoveries.log                 | Client's recoveries.log file records information about file restores that are performed on this client.                                                                                                                                                     |
| ~ebadmin/ <i>client-name</i> /eb_ci_data                     | Contains shell script values that are required by various shell scripts such as <b>startfind</b> .                                                                                                                                                          |
| /usr/epoch/man or /usr/local/man (or an alternate directory) | Contains the client's man pages.                                                                                                                                                                                                                            |
| /usr/epoch/bin/ebcreport                                     | Enables networked clients to produce reports such as media and backup history reports. (See man page.)                                                                                                                                                      |
| /usr/epoch/bin/ebcrecover                                    | Allows a networked client to restore files, directories, and filesystems. (See man page.)                                                                                                                                                                   |
| /usr/epoch/bin/edmcrestore                                   | Starts the EMC Data Manager Restore window from a networked client.                                                                                                                                                                                         |
| /usr/epoch/EB_DB                                             | Contains files for backups of database files.                                                                                                                                                                                                               |

---

## Volume Management Directory Structure

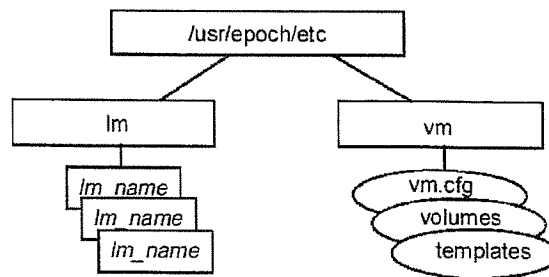
The `/usr/epoch/etc` directory has many subdirectories that contain vital EDM software database files. Two of these are the Library Manager and Volume Manager files. Figure A-11 shows these two subdirectories of `/usr/epoch/etc`.

---

Figure A-11

---

### `/usr/epoch/etc` Directory



---

## Library Manager Subdirectories

Each Library Manager resides in an individual subdirectory in `/usr/epoch/etc/lm`. For more information, see “Library Manager Configuration Files” on page C-9. The directory name matches the Library Manager name that appears in the Display area of the main volume management window.

Within each Library Manager directory, **lmconfig** creates several files. During start-up, the Library Manager reads the files in its directory to initialize the library unit it is managing and to set up its internal data structures. Table A-3 describes the files that are located in each of the Library Manager directories.

Table A-3

Files in `/usr/epoch/etc/lm/lm_name`

| File                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>lm.cfg</code>            | Configuration file for the Library Manager. Library Manager configuration includes parameters that define the hardware address (SCSI bus, target ID, and lun) of the device, Library Manager name, number of drives, and scheduling parameters for the robot and drive(s). See "Library Manager Configuration Files" on page C-9.                                                                                                                                                                                                                                                                     |
| <code>volid.dat</code>         | Inventory list of the library unit's contents. This file enables the Library Manager to start up without taking a complete inventory of the library unit. If <code>volid.dat</code> does not exist, the Library Manager inventories the library unit and creates the file. The Library Manager updates the <code>volid.dat</code> file when it: <ul style="list-style-type: none"> <li>• writes a label to a volume</li> <li>• moves a volume into or out of an inlet, slot, or drive</li> <li>• enables or disables a library unit slot</li> <li>• takes an inventory of the library unit</li> </ul> |
| <code>dn</code>                | Soft link to the physical name of the drive that is located in <code>/devices</code> . <i>n</i> indicates the drive number.                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <code>lm_in_drive_n.dat</code> | Drive contents file for each drive installed in the library unit; where <i>n</i> indicates the drive number.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <code>clog</code>              | Circular log file that contains a detailed description of the activity for this Library Manager.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <code>lmd</code>               | Link to the Library Manager executable daemon located in <code>/usr/epoch/lib/rvm</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>liblm_tnumber</code>     | Contains a number used internally to identify the Library Manager.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>lm_is_open</code>        | Lock file that ensures that only one copy of the Library Manager daemon is running at one time.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

---

**Volume Manager Files**

The directory `/usr/epoch/etc/vm` contains all files that are related to the Volume Manager. For more information, see “Volume Manager Configuration File” on page C-2. Table A-4 describes the files that are contained in `/usr/epoch/etc/vm`.

---

**Table A-4****Files in `/usr/epoch/etc/vm`**

---

| File Name                             | Description                                                                                                                                                                                                                                                             |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>clog</code>                     | Volume Manager daemon's circular log (clog) file. An ascii file that serves as a debug log and contains a detailed description of the <code>vmdaemon</code> 's activity.                                                                                                |
| <code>notd</code>                     | Link to the notify daemon located in <code>/usr/epoch/lib/rvm</code> .                                                                                                                                                                                                  |
| <code>notd.pid</code>                 | Notify daemon's process ID file.                                                                                                                                                                                                                                        |
| <code>notd_clog</code>                | Notify daemon's circular log (clog) file. An ascii file that serves as a debug log for the notify daemon. It contains a detailed description of the <code>notd</code> 's activity.                                                                                      |
| <code>templates</code>                | Template catalog contains volume templates for labeling media.                                                                                                                                                                                                          |
| <code>templates.def</code>            | Template record definition file.                                                                                                                                                                                                                                        |
| <code>templates.templateid.ndx</code> | Template index file.                                                                                                                                                                                                                                                    |
| <code>vm.cfg</code>                   | Volume Manager configuration file. This file contains parameters that define the location of the <code>vmdaemon</code> , set the message-logging level, and lists the Library Managers configured for this server. See “Volume Manager Configuration File” on page C-2. |
| <code>vm_is_open</code>               | Lock file which prevents two Volume Manager daemons from running at one time.                                                                                                                                                                                           |
| <code>vmdaemon</code>                 | Link to the Volume Manager executable daemon that is located in <code>/usr/epoch/bin</code> .                                                                                                                                                                           |
| <code>vmemd</code>                    | Link to the Volume Manager's erasing daemon that is located in <code>/usr/epoch/lib/rvm</code> .                                                                                                                                                                        |
| <code>volumes</code>                  | Volume catalog, contains complete volume information for the server.                                                                                                                                                                                                    |
| <code>volumes.def</code>              | Template catalog contains volume templates for labeling media.                                                                                                                                                                                                          |
| <code>volumes.volid.ndx</code>        | Template index file; an internal file used exclusively by the <code>vmdaemon</code> .                                                                                                                                                                                   |



---

## HSM Directory Structure

The `/usr/epoch/etc/mal` directory includes several subdirectories which contain server and client configuration and database files. This section provides detailed information on this directory structure.

---

## HSM Configuration Database

Every EDM HSM server and every migration client contains a migration configuration database. This database consists of structured text files that are updated by the **emstconf**, **emfsconf**, and **emsysconf** commands and by the functions you perform when using the EDM HSM Configuration window.

The database contains information that specifies which filesystems are stageable, when files should be staged, and the staging templates to which filesystems are assigned.

**CAUTION:** Although these files are text files, you should never attempt to modify them with an ordinary editor. The configuration commands and the EDM Backup Configuration window do more than just modify the files; they also know how to interact correctly with any staging processes that are running.

On client systems, the database also lists the fileserver and store to which staging templates are assigned.

The text files are stored in the `/usr/epoch/etc/mal/` directory.

Figure A-12

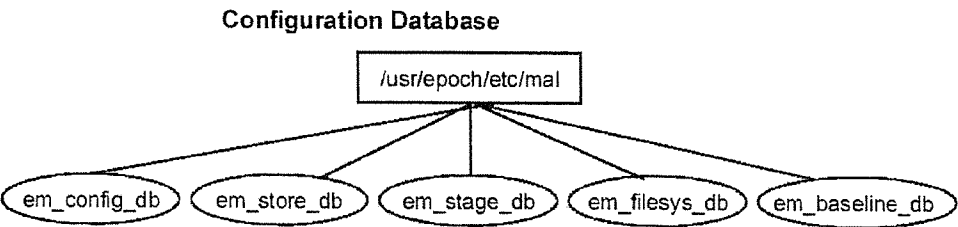


Table A-5 lists the database files.

**Table A-5** Migration Database Files

| Argument       | Description                                                                                                                                                                                            |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| em_config_db   | Text file that contains system-wide configuration data. You update this file with the <b>emsysconf</b> command or when you change global properties with the Configuration Interface.                  |
| em_stage_db    | Text file that contains staging template configuration data. You update this file with the <b>emstconf</b> command or when you change staging template information with the Configuration Interface.   |
| em_store_db    | Text file that contains client store information. You update this file with the <b>emstconf</b> command or when you change store information with the Configuration Interface.                         |
| em_filesys_db  | Text file that contains per-filesystem configuration data. You update this file when you issue the <b>emfsconf</b> command or when you change filesystem information with the Configuration Interface. |
| em_baseline_db | Text file that contains baseline backup information. This data is manipulated by the HSM software on behalf of the backup software.                                                                    |

**Network HSM Server Database**

The network migration server has a global configuration database that contains information about network migration activity and the default client store values. The global configuration database files reside in /usr/epoch/etc/mal.

**CAUTION:** Editing these files directly may result in loss of data.

Although both the emsd\_conf\_db and the emsd\_store\_db files contain editable text descriptions of the configuration, do not edit these files directly. Instead use the server's configuration commands or the EDM Backup Configuration window to make any modifications to the database. There are five database files.

Figure A-13

Global Database Files

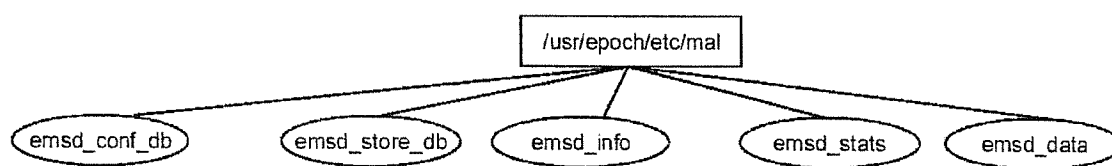


Table A-6

Global Database Files

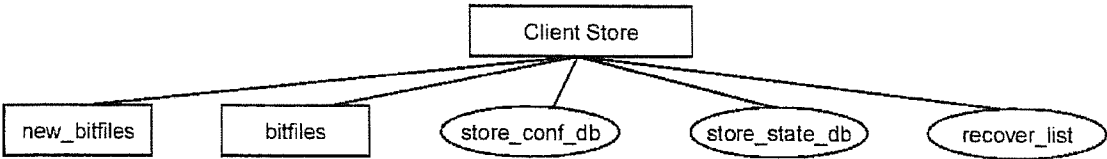
| File          | Description                                                                                                                     |
|---------------|---------------------------------------------------------------------------------------------------------------------------------|
| emsd_conf_db  | Text file that defines the limits on the HSM software protocol requests and the default values for client store configurations. |
| emsd_info     | Binary file that contains information about the currently executing emsd process.                                               |
| emsd_store_db | Text file with a list of configured client stores and their locations on the server.                                            |
| emsd_stats    | Binary file that contains cumulative statistics on HSM software protocol traffic and client agent activity.                     |
| emsd_data     | Binary file that contains the HSM usage history on the server.                                                                  |

## Client Stores

Each client store has its own file hierarchy and is logically independent from every other client store. The client store's top-level directory contains three files and two subdirectories.

As system administrator you see these files and directories when you list the contents of the client store directory.

**Figure A-14** Client Store Organization



The client store's top-level files and directories are listed in Table A-7.

**Table A-7** Client Store Files and Directories

| File/Directory | Description                                                                                                  |
|----------------|--------------------------------------------------------------------------------------------------------------|
| store_conf_db  | Text file of the store's configuration information.                                                          |
| store_state_db | Text file of the store's state information that the client agent keeps current.                              |
| recover_list   | List of the bitfiles to restore from the EDM server's backups.                                               |
| new_bitfiles   | Temporary holding directory for bitfiles that are being created as part of a stage out from a client system. |
| bitfiles       | Directory that contains the completed bitfiles in a 3-level hierarchy.                                       |

When the client agent creates a bitfile, it gives the bitfile a 16-digit hexadecimal name and places it in the new\_bitfiles directory. The bitfile remains there until it is completely written. Once the bitfile is complete, the client agent moves it to the bitfiles directory.

## Bitfiles

Bitfiles reside at the bottom layer of a directory hierarchy as shown in Figure A-15. Bitfile names are 16-digit hexadecimal numbers representing the lower 64-bits of a file's bitfile ID. (The bitfile ID consists of the bitfile name plus the store ID.) Migration uses this organization so that a bitfile can be located by using the hexadecimal encoding of the bitfile ID.

Figure A-15

Bitfile Hierarchy

